世界知的所有権機関
国際事務局

PCT

特許協力条約に基づいて公開された国際出願

(51) 国際特許分類6
G09C 1/00, H04L 9/06

A1

(11) 国際公開番号 WO99/38143

(43) 国際公開日 1999年7月29日(29.07.99)

(21) 国際出願番号 PCT/JP99/00337

(22) 国際出願日 1999年1月27日(27.01.99)

(30) 優先権データ
特願平10/13572 1998年1月27日(27.01.98) JP
特願平10/13573 1998年1月27日(27.01.98) JP
特願平10/147479 1998年5月28日(28.05.98) JP

(71) 出願人（米国を除くすべての指定国について）
日本電信電話株式会社(NIPPON TELEGRAPH AND TELEPHONE CORPORATION)[JP/JP]
〒163-8019 東京都新宿区西新宿三丁目19番2号 Tokyo, (JP)
(72) 発明者；および
(75) 発明者／出願人（米国についてのみ）
神田雅透(KANDA, Masayuki)[JP/JP]
高嶋洋一(TAKASHIMA, Youichi)[JP/JP]
青木和麻呂(AOKI, Kazumaro)[JP/JP]
植田広樹(UEDA, Hiroki)[JP/JP]
太田和夫(OHTA, Kazuo)[JP/JP]
〒163-1419 東京都新宿区西新宿3丁目20-2
日本電信電話株式会社内 Tokyo, (JP)

松本 勉(MATSUMOTO, Tsutomu)[JP/JP]
〒227-0048 神奈川県横浜市青葉区柿の木台13-45
Kanagawa, (JP)
(74) 代理人
弁理士 草野 卓, 外(KUSANO, Takashi et al.)
〒160-0022 東京都新宿区新宿四丁目2番21号
相模ビル Tokyo, (JP)

(81) 指定国 CA, US, 欧州特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE)

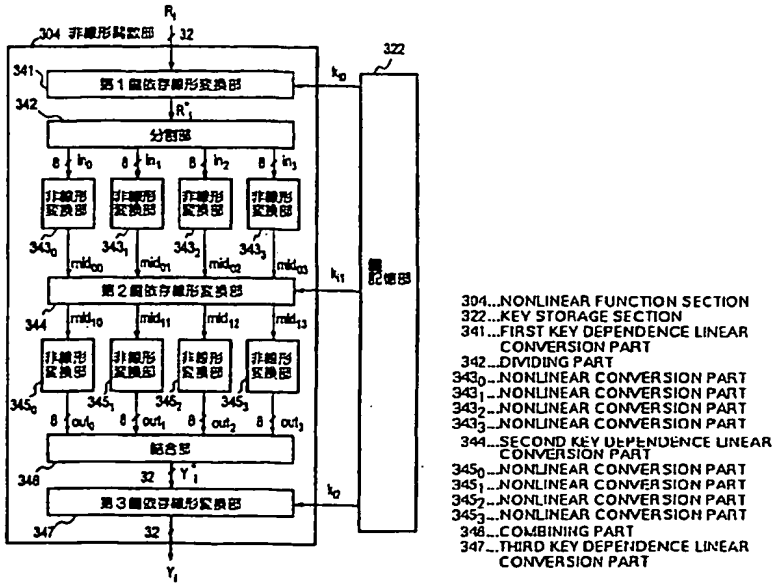
添付公開書類
国際調査報告書

54)Title: DATA CONVERTER AND RECORDING MEDIUM ON WHICH PROGRAM FOR EXECUTING DATA CONVERSION IS RECORDED

54)発明の名称 データ変換装置及びそれを実施するプログラムが記録された記録媒体

57) Abstract

A data converter has rounding units (38) each including a nonlinear function section 304), and each nonlinear function section (304) ncludes a first key dependence linear conversion part (341) for performing linear conversion based on a subkey, a dividing part (342) for dividing he output of the first key dependence linear conversion part (341) into n pieces of sub-data, irst nonlinear conversion parts (343) for nonlinearly converting the sub-data, a second key lependence linear conversion part (344) for inearly converting the nonlinear conversion outputs based on a subkey and outputting n pieces of conversion sub-data, second nonlinear conversion parts (345) for nonlinearly converting he conversion sub-data, and a combining part 346) for combining the nonlinear conversion outputs and producing the output of the nonlinear function section. An nxn matrix epresenting the linear conversion by the second key dependence linear conversion part (344) is composed of n vectors such that the hamming weight for a safety threshold T is T-1 or more. Therefore, the resistance to the difference decryption method and the linear decryption method is strengthened.



(57)要約

それぞれ非線形関数部(304)を含む複数のラウンド処理部(38)を有し、各非線形関数部(304)は、副鍵に基づいて線形変換を行う第１鍵依存線形変換部(341)と、その出力をｎ個のサブデータに分割する分割部(342)と、それらサブデータをそれぞれ非線形変換する第１非線形変換部(343)と、それらの非線形変換出力を副鍵に基づいて線形変換し、ｎ個の変換サブデータを出力する第２鍵依存線形変換部(344)と、それら変換サブデータを非線形変換する第２非線形変換部(345)と、その非線形変換出力を結合して非線形関数部の出力とする結合部(346)を含み、その第２鍵依存線形変換部(344)の線形変換処理を表すｎ×ｎ行列を、安全性閾値Ｔに対しハミング重みがT-1 以上となるｎ個のベクトルにより構成することにより差分解読法、線形解読法に対し耐性を強くする。

PCTに基づいて公開される国際出願のパンフレット第一頁に掲載されたPCT加盟国を同定するために使用されるコード(参考情報)

AE	アラブ首長国連邦	ES	スペイン	LI	リヒテンシュタイン	SG	シンガポール
AL	アルバニア	FI	フィンランド	LK	スリ・ランカ	SI	スロヴェニア
AM	アルメニア	FR	フランス	LR	リベリア	SK	スロヴァキア
AT	オーストリア	GA	ガボン	LS	レソト	SL	シエラ・レオネ
AU	オーストラリア	GB	英国	LT	リトアニア	SN	セネガル
AZ	アゼルバイジャン	GD	グレナダ	LU	ルクセンブルグ	SZ	スワジランド
BA	ボズニア・ヘルツェゴビナ	GE	グルジア	LV	ラトヴィア	TD	チャード
BB	バルバドス	GH	ガーナ	MC	モナコ	TG	トーゴー
BE	ベルギー	GM	ガンビア	MD	モルドヴァ	TJ	タジキスタン
BF	ブルギナ・ファソ	GN	ギニア	MG	マダガスカル	TM	トルクメニスタン
BG	ブルガリア	GW	ギニア・ビサオ	MK	マケドニア旧ユーゴスラヴィア共和国	TR	トルコ
BJ	ベナン	GR	ギリシャ	ML	マリ	TT	トリニダッド・トバゴ
BR	ブラジル	HR	クロアチア	MN	モンゴル	UA	ウクライナ
BY	ベラルーシ	HU	ハンガリー	MR	モーリタニア	UG	ウガンダ
CA	カナダ	ID	インドネシア	MW	マラウイ	US	米国
CF	中央アフリカ	IE	アイルランド	MX	メキシコ	UZ	ウズベキスタン
CG	コンゴー	IL	イスラエル	NE	ニジェール	VN	ヴィェトナム
CH	スイス	IN	インド	NL	オランダ	YU	ユーゴースラビア
CI	コートジボアール	IS	アイスランド	NO	ノールウェー	ZA	南アフリカ共和国
CM	カメルーン	IT	イタリア	NZ	ニュー・ジーランド	ZW	ジンバブエ
CN	中国	JP	日本	PL	ポーランド		
CU	キューバ	KE	ケニア	PT	ポルトガル		
CY	キプロス	KG	キルギスタン	RO	ルーマニア		
CZ	チェッコ	KP	北朝鮮	RU	ロシア		
DE	ドイツ	KR	韓国	SD	スーダン		
DK	デンマーク	KZ	カザフスタン	SE	スウェーデン		
EE	エストニア	LC	セントルシア				

明細書

データ変換装置及びそれを実施するプログラムが記録された記録媒体

技術分野

この発明は、データの通信または蓄積において、データを秘匿するための暗号化装置に用いられるデータ変換装置、特に、秘密鍵の制御のもとでデータをブロック単位で暗号化または復号を行う共通鍵暗号方式による暗号化装置に適するデータ変換装置及びそのデータ変換装置により実行するプログラムを記録した記録媒体に関する。

従来の技術

高速かつ安全な共通鍵暗号を構成するために、暗号化対象のデータを適当な長さのブロックに分割し、そのデータブロック毎に暗号化する方法をブロック暗号と呼ぶ。通常、ブロック暗号は、暗号化の対象である入力データを攪乱するためのデータ拡散部と、暗号装置に入力された秘密の共通鍵（以降ではこの鍵を主鍵と呼ぶ）を入力としてデータ拡散部が利用する一連の副鍵を生成するための鍵スケジュール部とから構成されている。データを秘匿するためのこのようなデータ変換装置に使用される代表的な共通鍵暗号方式には、米国連邦標準暗号であるＤＥＳ（Data Encryption Standard）暗号がある。

図１は、ＤＥＳ暗号の機能構成を示す。ＤＥＳ暗号では、６４ビットの秘密鍵（うち８ビットはパリティに用いられる）を用い、６４ビットのデータブロック単位に暗号化または復号を行う。図１において、暗号化処理は、データ拡散部１０において平文Mの６４ビットを初期転置部１１において初期転置で変換した後、３２ビットごとのブロックデータL_0, R_0に分割される。次に、ブロックデータR_0は図２の第ｉラウンド処理部１４$_i$（i=0, 1, …, 15）に示すデータ変換部である関数演算部（ラウンド関数とも呼ばれる）１２へ入力され、４８ビットの副鍵k_0の制御のもとに$f(R_0, k_0)$に変換される。この変換データ$f(R_0, k_0)$とブロックデータL_0との排他的論理和をＸＯＲ回路１３でとり、更に、その出力値とブロックデータR_0とを入れ替えて、次のブロックデータL_1, R_1とする。即ち、

$$R_1 = L_0 \oplus f(R_0, k_0)$$

$L_1 = R_0$

である。ここで⊕は排他的論理和を表わすものとする。このように２つのブロックデータL_0, R_0を入力として演算部１２と排他的論理和回路１３とデータの入れ替え（スワップ）とによりL_1, R_1を出力する第０段ラウンド処理部１４$_0$が構成され、同じようなラウンド処理部14_1～14_{15}が縦続的に設けられる。第ｉ段ラウンド処理部１４$_i$による処理を第ｉ段のラウンド処理と呼ぶことにする。ただし、i=0, 1, …, 15 である。つまり各ラウンド処理部１４$_i$（０≦ｉ≦１５）では、

$R_{i+1} = L_i \oplus f(R_i, k_i)$

$L_{i+1} = R_i$

の処理が行われ、最後にR_{16}, L_{16}を統合して６４ビットにした後、最終転置部１５において最終転置で変換して暗号文６４ビットを出力する。なお、最終転置部１５は初期転置部１１の逆変換に相当する。

復号処理においては、関数ｆ（関数演算部１２）に入力する副鍵k_0, k_1, …, k_{14}, k_{15}の順序だけを逆転させて、k_{15}, k_{14}, …, k_1, k_0の順に入力するようにする点を除けば、暗号化処理と同じ手順で実行できる。その場合、最終段ラウンド処理部14_{15}のスワップ出力L_{16}, R_{16}を、図に示すように更にスワップするように構成することにより、復号処理において暗号文を初期転置１１に入力して図１の処理を実行することにより、最終転置１５の出力に平文がそのまま得られる。鍵スケジュール部２０は、拡大鍵生成ルーチン１６で６４ビットの主鍵から８ビット分のパリティビットを除いた５６ビットを左右２８ビットずつの鍵データに分割し、それら各２８ビットの左右鍵データに対し１６段の並び替え処理を行い、各段の並び替え処理結果の左右鍵データ（計５６ビット）を縮約転置によりデータ拡散部１０の対応するラウンドに与える１６個の４８ビットの副鍵k_0, k_1, …, k_{14}, k_{15}として生成する。

関数演算内部１２の処理は、図２に示すように行われる。まず、３２ビットのブロックデータR_iは拡大転置部１７で４８ビットデータ$E(R_i)$に変換される。これに副鍵k_iとで排他的論理和を XOR回路１８で取り、４８ビットデータ$E(R_i) \oplus k_i$に変換した後、８個の６ビットごとのサブブロックデータに分割する。この８個

のサブブロックデータはそれぞれ異なるS-box S_0～S_7に入力され、各々が４ビットの出力を得る。なお、このS-box S_j (j=0, 1, …, 7) は６ビットの入力データから４ビットの出力データに変換する非線形変換テーブルであり、ＤＥＳ暗号の本質的な安全性を担っている部分である。S-box S_0～S_7の８つの出力データは、再び連結されて３２ビットデータになった後、転置変換部１９を経て、図２に示されるように関数演算部１２の出力$f(R_i, k_i)$となる。この出力は、図１に示されるように、L_iと排他的論理和されてR_{i+1}となる。

次に、暗号解読法について述べる。ＤＥＳ暗号を始めとする従来の共通鍵暗号方式についてはさまざまな方面から暗号解読が試みられており、そのなかでも、極めて効果的な解読法は、E. Biham およびA. Shamirによって "Differential Cryptanalysis of DES-like Cryptosystems," Journal of Cryptology, Vol.4, No. 1, pp.3-72に提案された差分解読法と、松井によって "Linear Cryptanalysis Method for DES cipher," Advances in Cryptology-EUROCRYPT'93(Lecture Notes in Computer Science 765), pp.386-397 に提案された線形解読法である。

ＤＥＳ暗号に対する差分解読法は、２つのデータX, X*の差分を

$$\Delta X = X \oplus X^*$$

としたとき、解読者が入手している平文・暗号文の２組を以下の式に適用して、最終ラウンド14_{15}における副鍵k_{15}を求めることを目的としている。図１の暗号化処理において、第１及び第２平文を入力したときの各ラウンド処理部１４$_i$での入力ブロックデータを(L_i, R_i), (L^*_i, R^*_i)とする。上記差分の定義により次式

$$\Delta L_i = L_i \oplus L^*_i$$

$$\Delta R_i = R_i \oplus R^*_i$$

が成立する。図１において、$L_{15}=R_{14}$, $L^*_{15}=R^*_{14}$, $L_{16}=R_{15}$, $L^*_{16}=R^*_{15}$なので次式

$$R_{16} = L_{15} \oplus f(R_{15}, k_{15})$$

$$R^*_{16} = L^*_{15} \oplus f(R^*_{15}, k_{15})$$

が成立し、これら２つの式の両辺の排他的論理和をとると次式

$$\Delta R_{16} = \Delta L_{15} \oplus f(L_{16}, k_{15}) \oplus f(L_{16} \oplus \Delta L_{16}, k_{15})$$

が得られ、その両辺と$\Delta R_{14}=\Delta L_{15}$との排他的論理和をとることにより次の式が得ら

れる：

$$f(L_{16},\ k_{15}) \oplus f(L_{16} \oplus \Delta L_{16},\ k_{15}) = \Delta R_{16} \oplus \Delta R_{14}$$

このとき、L_{16}、ΔL_{16}、ΔR_{16} は暗号文から得られるデータであるので入手済みの情報である。このため、解読者が ΔR_{14} を正しく求めることができるならば、上式は k_{15} のみが未知定数となり、入手済みの平文・暗号文の組を用いて k_{15} に関する全数探索を行うことで、解読者は必ず正しい k_{15} を見つけだすことができる。従って、副鍵 k_{15} が求まってしまえば、残り８(即ち56-48) ビットは総当たりでも簡単に求まってしまう。

一方、ΔR_{14} についてみてみると、この値は中間差分値であるため、一般には求めることが困難である。そこで、第０ラウンドから最終ラウンドの一つ前までの第１４段ラウンドまでにおいて、各ラウンド（段）が確率 p_i で次式

$$\Delta R_{i+1} = \Delta L_i \oplus \Delta\{f(\Delta R_i)\}$$

$$\Delta L_{i+1} = \Delta R_{i+1}$$

のように近似されるとおく。ここでのポイントは、ある ΔR_i が入力されたとき、副鍵 k_i の値に関わらず、確率 p_i で $\Delta\{f(\Delta R_i)\}$ を予測できるということにある。このように近似できるのは、非線形な変換であるS-box において、入力差分によっては差分出力の分布に極めて大きな偏りが生じるためである。例えば、S-box S_0 では、入力差分"$110100_{(2)}$"のとき、1/4 の確率で出力差分"$0010_{(2)}$"に変換されるためである。そこで、各々のS-box が確率 p_{si} で入力差分と出力差分との関係が予測できるとおき、これらを組み合わせることで各ラウンドの近似を求める。更に、各ラウンドでの近似を連結していくことで、ΔR_{14} は確率 $P=\Pi_{i=0}^{13} p_i$ で ΔL_0、ΔR_0（ΔL_0、ΔR_0 は平文から得られるデータであるので入手済みの情報である）から求められることになる。なお、この確率Ｐが大きいほど、暗号解読が容易である。このようにして、副鍵 k_{15} が求められると、今度は１段少ない１５段ＤＥＳ暗号とみなして、同様の手法で、副鍵 k_{14} を求めていくということを繰り返して、最終的に副鍵 k_0 まで求めていく。

この解読法による暗号解読が成功するかどうかは確率Ｐに依存し、この値が大きいほど成功しやすい。Biham らによると、この解読法では、2^{47}組の解読者が選

択した既知平文・暗号文の組を入手できればＤＥＳ暗号を解読できるとしている。

また、ＤＥＳ暗号に対する線形解読法は、以下の線形近似式を構成し、解読者が入手している平文・暗号文の組による最尤法を用いて副鍵を求めることを目的としている。

$$(L_0,\ R_0)\,\Gamma\,(L_0,\ R_0)\oplus(L_{16},\ R_{16})\,\Gamma\,(L_{16},\ R_{16})$$
$$=(k_0,\ k_1,\ \cdots,\ k_{15})\,\Gamma\,(k_0,\ k_1,\ \cdots,\ k_{15})$$

ただし、Γ(X) はXの特定のビット位置を選択するベクトルを表し、マスク値という。

線形近似式の役割は、暗号アルゴリズム内部を線形表現で近似的に置き換え、平文・暗号文の組に関する部分と副鍵に関する部分とに分離することにある。つまり、平文・暗号文の組に関して、平文の特定のビット位置の値と暗号文の特定のビット位置の値との全ての排他的論理和が一定値となり、その値は副鍵の特定のビット位置の値の排他的論理和に等しくなることを表している。従って、解読者は

$$(L_0,\ R_0)\,\Gamma\,(L_0,\ R_0)\oplus(L_{16},\ R_{16})\,\Gamma\,(L_{16},\ R_{16})$$

の情報から

$$(k_0,\ k_1,\ \cdots,\ k_{15})\,\Gamma\,(k_0,\ k_1,\ \cdots,\ k_{15})\ (1\text{ビット})$$

の情報が得られるということになる。このとき、$(L_0,\ R_0)$, $(L_{16},\ R_{16})$はそれぞれ平文・暗号文のデータであるので入手済みの情報である。このため、解読者が$\Gamma(L_0,\ R_0)$, $\Gamma(L_{16},\ R_{16})$, $\Gamma(k_0,\ k_1,\ \cdots,\ k_{15})$ を正しく求めることができるならば、$(k_0,\ k_1,\ \cdots,\ k_{15})\,\Gamma\,(k_0,\ k_1,\ \cdots,\ k_{15})$ (１ビット) を求めることができる。

ＤＥＳ暗号では、非線形な変換が起きる部分はS-box しかないため、S-box についてのみ線形表現ができれば、容易に線形近似式が構成できる。そこで、各々のS-box が確率p_{si} で線形表現できるとおく。ここでのポイントは、S-box に対する入力マスク値が与えられたとき、確率p_{si} でその出力マスク値を予測できるということにある。これは、非線形変換テーブルであるS-box において、入力マスク値によっては差分マスク値の分布に極めて大きな偏りが生じるためにおこる。例えば、S-box S_4では、入力マスク値"$010000_{(2)}$"のとき、3/16の確率で出力マスク値

"$1111_{(2)}$"が予測されるためである。これらS-box におけるマスク値を組み合わせることによって、各ラウンドが確率p_iで入力マスク値と出力マスク値との線形表現ができ、各ラウンドでの線形表現を連結していくことで、$\Gamma(L_0, R_0)$, $\Gamma(L_{16}, R_{16})$, $\Gamma(k_0, k_1, \cdots, k_{15})$ は確率

$$P = 1/2 + 2^{15}\Pi_{i=0}^{15} | p_i - 1/2 |$$

で求められることになる。なお、この確率Ｐが大きいほど、暗号解読が容易である。

松井によると、この解読法で、2^{43}組の既知平文・暗号文の組を用いて、ＤＥＳ暗号の解読に成功している。

さて、上記の解読法に対抗するためには、確率Ｐが十分に小さくなればよい。このため、確率Ｐを小さくするための提案がさまざま行われており、なかでも従来の暗号方式において、もっとも簡単に安全性を高めるための方法がラウンド数（段数）を増やすことであった。例えば、ＤＥＳ暗号を３つつなげたTriple-DES暗号は、実質的にＤＥＳのラウンド数を１６段から４８段に増やした暗号方式であり、確率Ｐは、ＤＥＳ暗号よりもはるかに小さい。

しかし、上記の解読法に対抗するための対策として、ラウンド数（段数）を増加させることは、暗号化速度を犠牲にすることになる。例えば、ラウンド数を３倍に増やせば、暗号化速度は1/3 になる。つまり、現在のＤＥＳ暗号の暗号化速度はPentium PCクラスで約１０Mbpsであるため、Triple－DES 暗号ともなると約3.5Mbps まで暗号化速度が低下する。一方で、ネットワークやコンピュータなどは年々高速化しており、データ変換装置もそれらの高速化に対応したものが望まれている。このため、従来のデータ変換装置では、それらの高速化の要求に対して、安全性と高速性を同時に満たすことはきわめて困難な状況になっている。

また、上述のように、差分解読法や線形解読法では最終ラウンドの副鍵が求められる。ＤＥＳ暗号の場合、最終ラウンドの副鍵が求められてしまうと、そこから容易に主鍵が求められてしまうという欠点があるため、鍵スケジュール部２０の副鍵と主鍵の対応関係を更に複雑にすることで、安全性を向上する方法が米国特許No. 4, 850, 019に提案されている。その原理的構成を図３に示す。上記米国特

許では、データ拡散部(fk)を用いて主鍵から副鍵を生成することで、例えばデータ拡散部(fk)を用いて主鍵から副鍵を生成することで、例えば副鍵が求まったとしても簡単には主鍵が求まらないことを期待している。

上記米国特許で示されている鍵スケジュール部２０の概要を図３を参照して次に説明する。拡大鍵生成ルーチン21はそれぞれ鍵拡散部22_0～$22_{N/2-1}$を有するN/2(例えばN=16)段の鍵処理部21_0～$21_{N/2-1}$から成る。各鍵処理部21_j(j=0, 1, …, N/2-1)は与えられた各３２ビットの入力左右鍵データを拡散処理し、その出力左右鍵データは左右が入れ替えられて次段の鍵処理部21_{j+1}に左右鍵データとして入力される。初段を除く各鍵処理部21_jは排他的論理和部23_jを有し、前段の鍵処理部21_{j-1}の入力左鍵データと出力左データの排他的論理和を演算し、その演算結果を鍵拡散部22_jに与える。鍵処理部21_jの入力左鍵データは鍵拡散部22_jで排他的論理和23_jの出力により拡散され次段の入力右鍵データとして出力され、鍵処理部21_jの入力右鍵データは次段の入力左鍵データとして出力される。各鍵拡散部22_jの出力はビット分割されて２つの副鍵Q_{2j}, Q_{2j+1}(即ちk_i, k_{i+1})として図１の対応する第i=2jラウンド処理段と第i+1=2j+1ラウンド処理段に与えられる。

６４ビットの主鍵はそれぞれ３２ビットの左右鍵データに分割され、初段の鍵処理部21_0において左鍵データを右鍵データで鍵拡散部22_0により拡散して拡散左鍵データとし、この拡散左鍵データと右鍵データの左右を入れ替えて左右鍵データとして次段の鍵処理部21_1に与える。鍵処理部21_0～$21_{N/2-1}$の鍵拡散部22_0～$22_{N/2-1}$の出力は副鍵k_0～k_{N-1}として図１に示したデータ拡散部１０の対応するラウンド14_0～14_{N-1}に与えられる。

しかしながら、図３の拡大鍵生成ルーチン21において、各鍵拡散部22_jは２つの入力データから１対の鍵データ（副鍵Q_{2j}, Q_{2j+1}）を生成出力する関数であり、これら２つの入力データの一方と、出力データとが判明すれば、他方の入力データを知ることができる性質を備えているとき、仮に３対の副鍵Q_{2j-2}, Q_{2j-1}; Q_{2j}, Q_{2j+1}; Q_{2j+2}, Q_{2j+3}が判明したと仮定すると、第j+1段の鍵拡散部22_{j+1}において、その出力（副鍵Q_{2j+2}, Q_{2j+3}）と、一方の入力データ（副鍵Q_{2j-2}, Q_{2j-1}）がわかっているので、他の入力データ（即ち排他的論理和部23_{j+1}の出力データ）を求めることができ、その

求めたデータと排他的論理和部23_{j-1}の一方の入力データである副鍵Q_{2j}, Q_{2j-1} とから排他的論理和部23_{j-1}の他方の入力データである前段（第ｊ段）の鍵拡散部22_jの入力、即ち、３段前（第j-2段）の鍵拡散部22_{j-2}の出力である副鍵Q_{2j-1}, Q_{2j-3} が求まる。この様な操作を順次繰り返すことで、データ拡散部１０におけるデータを解析すること無しに、鍵スケジュール部２０におけるデータのみを解析することで全ての副鍵を決定できてしまう。ここでは連続した３段分の副鍵が求まると全ての副鍵が求まることを指摘したが、連続２段分の副鍵を知っている場合には、隣接する他の１段分の副鍵を総当たりで推定することでも、攻撃に成功する。

図１のラウンド処理の最終段をi=N とすると、差分解読法及び線形解読法では、副鍵k_N, k_{N-1}が求まりやすい。それらを使って上述のように鍵スケジュール部21における鍵データの解析を行うことにより全ての副鍵が求まる可能性がある。

この発明の第１の目的は、ラウンド関数ｆ（関数演算部）を安全性と高速性を同時に満たすような構造にすることによって、ラウンド数（段数）を大幅に増加させることなく安全性を確保し、かつ高速な暗号化処理が可能となるようなデータ変換装置及びデータ変換を実施するプログラムを記録した記録媒体を提供することにある。

この発明の第２の目的は、一部の副鍵が知られた場合においても、鍵スケジュール部を解析するだけでは他の副鍵及び主鍵が簡単には求まらないような鍵スケジュール部を実現することである。

発明の開示

この発明の第１の目的を達成するため、特に非線形関数部において、非線形関数部の入力データに対し鍵記憶部に蓄積された第１鍵データに基づいて線形変換を行う第１鍵依存線形変換部と、その第１鍵依存線形部の出力データをｎ個のサブデータに分割する分割部と、これらの各サブデータに非線形変換を行う第１非線形変換部と、その第１非線形変換部の各々の出力サブデータに対し第２鍵データに基づいて線形変換を行う第２鍵依存線形変換部と、その第２鍵依存線形変換部の出力サブデータに非線形変換を行う第２非線形変換部と、その第２非線形変換部の出力サブブロックを結合して非線形関数部の出力データとする結合部とを

備えており、上記第２鍵依存線形変換手段は、その入力に対しｎ×ｎ行列で規定される排他的論理和を行う線形変換部を含んでいることを特徴とする。

この発明によれば、第１及び第２非線形変換部における差分確率・線形確率がｐ（＜１）であるとき、各段を近似するときの差分確率・線形確率は$p_i \leqq p^2$（ただし、差分解読法では関数ｆ（非線形関数部）への入力差分が０でないとき、線形解読法では関数ｆでの出力マスク値が０でないとき）となることが保証される。また、関数ｆが全単射であるとき、暗号装置の段数を３ｒとすると、暗号装置としての確率は$P \leqq p_i^{2r} \leqq p^{4r}$となる。更に、特にｎ＝４の場合の第２鍵依存線形変換部が４つのサブデータから選んだ３つと鍵データの４分割の１つの排他的論理和の全ての組合せを演算する構造を有するならば、各段を近似するときの確率は$p_i \leqq p^4$、暗号装置としての確率は$P \leqq p_i^{2r} \leqq p^{8r}$となる。ｎ＝８の場合の第２鍵依存線形変換部が８つのサブデータから選んだ６つ又は５つと、鍵データの８分割の１つとの排他的論理和の全ての組み合わせを演算する構造を有するならば、格段を近似するときの確率は$p_i \leqq p^5$となり、暗号装置としての確率は$P < p_i^{2r} < p^{10r}$となる。

また、第１および第２非線形変換部はそれぞれの手段が完全に並列処理できるような配置となっているため、高速化に寄与する。

ゆえに、差分解読法や線形解読法に対する安全性を有した高速な非線形関数を構成でき、安全性と高速性を両立させたデータ変換装置を提供することが可能になる。

この発明の第２の目的を達成するため、鍵スケジュール装置において、鍵拡散部（関数f_i）と同様の働きをするＧ関数部を用い、Ｇ関数部の出力であるＬ成分が記憶部に一旦記憶され、必要な個数だけＬ成分を求めた後に、それぞれのＬ成分から出来るだけ均一に必要となる情報を抽出して副鍵を生成するデータ抽出機能を備えたＨ関数部が設けられる。更に、Ｇ関数部の出力であるＬ成分からそれぞれの副鍵として使用される部分情報がＨ関数部で抽出され、記憶部に記憶され、必要な個数のＬ成分から部分情報を抽出することで副鍵が生成される。

図面の簡単な説明

図１は従来のＤＥＳ暗号装置の機能構成を示す図。

図２は図１中のｆ関数演算部１２の具体的機能構成を示す図。

図３は図２における拡大鍵生成ルーチン21の構成例を示す図。

図４はこの発明の第１実施例の機能構成を示す図。

図５は第１実施例における非線形関数部304 の詳細な機能構成例を示す図。

図６は図５における最適線形変換部を決めるための非線形関数部の基本構成図。

図７は図５中の第２鍵依存線形変換部347 の具体例を示す図。

図８Ａは第２実施例における非線形変換部343 の等価的機能構成を示す図。

図８Ｂは第２実施例における非線形変換部344 の等価的機能構成を示す図。

図８Ｃは第２実施例における非線形変換部345 の等価的機能構成を示す図。

図８Ｄは第２実施例における非線形変換部346 の等価的機能構成を示す図。

図９はこの発明の第２実施例における第２鍵依存線形変換部347 の機能構成を示す図。

図１０は実施例３における非線形関数部343 の機能構成を示す図。

図１１はデータ変換処理をコンピュータで実施する処理手順を示すフローチャート。

図１２は図１１におけるステップＳ３の詳細な処理手順を示すフローチャート。

図１３はこの発明の第４実施例の機能構成を示す図。

図１４は図１３における非線形関数部304 の機能構成を示す図。

図１５Ａは探索演算を削減するため限定した構成の線形変換部を示す図。

図１５Ｂは図１５Ａにおける１つの変換ボックスの構成を示す図。

図１６は探索アルゴリズムにより決定した線形変換部344Aの構成例を示す図。

図１７は第４実施例における図１４中の第２鍵依存線形変換部344 の機能構成例を示す図。

図１８は第４実施例における図１４中の第２鍵依存線形変換部344 の他の機能構成を示す図。

図１９は第４実施例における図１４中の第２鍵依存線形変換部344 の更に他の機能構成を示す図。

図２０Ａは第５実施例における非線形変換部343_n'の機能構成を示す図。

図２０Ｂは非線形変換部343_1'の機能構成を示す図。

図２０Ｃは非線形変換部343_7'の機能構成を示す図。

図２１は第５実施例における第２鍵依存線形変換部344 の機能構成を示す図。

図２２は記録媒体に記録されたデータ処理手順プログラムを実行する構成を示す図。

図２３Ａはこの発明による鍵生成スケジュールの原理機能構成を示すブロック図。

図２３Ｂはこの発明による他の鍵生成スケジュールの原理機能構成を示すブロック図。

図２４は図２３Ａ又は２３Ｂ中の中間鍵生成部230 の機能構成例を示すブロック図。

図２５はこの発明を図３の鍵スケジュール部に適用した場合の図２４中のＧ関数部２２の機能的構成を示すブロック図。

図２６はこの発明を図３の鍵スケジュール部に適用した場合の図２３Ａ中の副鍵生成部240 の機能的構成を示すブロック図。

図２７はこの発明を図３の鍵スケジュール部に適用した場合の図２３Ｂ中の副鍵生成部250 の機能的構成例を示すブロック図（この実施例では副鍵生成部がビット抽出機能をそなえたＨ関数部をその一部に含む）

図２８はこの発明を１２８ビットを１つのブロックとするFeistel型暗号に適用できるように応用した場合のＧ関数部22の機能的構成を示すブロック図。

発明を実施する最良の形態

第１実施例

以下、この発明の一実施例を図面を用いて説明する。

図４は、この発明の一実施例を示すデータ変換装置における、暗号化処理手順の機能構成を示したものである。データ変換装置は、データ拡散部１０と鍵スケジュール部２０とから構成され、この発明によるデータ変換装置においても、デ

ータ拡散部１０には入力データを左、右ブロックデータL_0, R_0に分割し、それらを順次ラウンド処理するＮ段に縦続接続されたラウンド処理部38_0～38_{N-1}が設けられ、各ラウンド処理部38_i (i=0, 1, …, N-1)は図１のラウンド関数部１２に対応する非線形関数部304 と、図１の XOR回路１３に対応する線形演算部305 と、交換部306 とから構成されている。

平文に相当する入力データＭを入力部301 から暗号装置内に入力する。鍵スケジュール部２０は、鍵入力部320 と、鍵データ生成部321 と、鍵記憶部322 とから構成され、予め、鍵入力部320 から入力されたデータ（主鍵Ｋ）に基づいて鍵データ生成部321 により複数の鍵データ（副鍵）

{fk; k_{00}, k_{01}, k_{02}; k_{10}, k_{11}, k_{12}; …; $k_{(N-1)0}, k_{(N-1)1}, k_{(N-1)2}$; ek}

が生成され、鍵記憶部322 に保持される。入力平文データＭは、鍵記憶部322 に蓄積されている鍵データfkによる初期鍵依存変換部302 で変換された後、初期分割部303 で左、右ブロックデータL_0, R_0に分割される。例えば６４ビットのデータが３２ビットずつのブロックデータL_0, R_0にビット分割される。初期鍵依存変換部302 では、例えば鍵データfkと入力データＭとの排他的論理和や鍵データfkによる入力データＭのビットローテーション（ビット回転）などの線形変換、もしくは乗算などを組み合わせた非線形変換が行われる。

右ブロックデータR_0は、鍵記憶部322 に蓄積されている鍵データk_{00}, k_{01}, k_{02}とともにこの発明により特徴的に構成された非線形関数部304 に入力され、非線形関数部304 で非線形変換処理によりデータY_0に変換される。データY_0と左ブロックデータL_0は線形演算部305 で線形演算されてデータL_0^*に変換される。データL_0^*とデータR_0は交換部306 でデータ位置の交換（スワップ）が行われ、$L_1 \leftarrow R_0$, $R_1 \leftarrow L_0^*$とされ、L_1, R_1が次の第１段ラウンド処理部38_1に入力される。

以下、第ｉ段ラウンド処理部38_i (i=0, 1, …, N-1)において、２つの入力ブロックデータL_i, R_iについて上記と同様の処理を繰り返し行う。即ち、左、右ブロックデータL_i, R_iについて、データR_iは、鍵記憶部322 に蓄積されている鍵データk_{i0}, k_{i1}, k_{i2}とともに非線形関数部304 に入力され、非線形関数部304 で非線形変換処理が行われて、データY_iに変換される。データY_iとデータL_iは線形演算部305 で演算さ

れてデータL_i^*に変換される。データL_i^*とデータR_iは交換部306でデータ位置の交換が行われ、$L_{i-1} \leftarrow R_i$, $R_{i-1} \leftarrow L_i^*$のように交換される。線形演算部305は例えば排他的論理和演算を行うものである。

暗号化を行うためのデータ変換装置としての安全性を確保するための適切な繰り返し回数（ラウンド回数）をNとすると、ラウンド処理部38_0～38_{N-1}による繰り返し処理の結果、左、右ブロックデータL_N, R_Nが得られる。このデータL_N, R_Nを最終結合部307で結合し、つまり例えば３２ビットの各L_N, R_Nをビット結合して６４ビットのデータとし、その後、鍵記憶部322に蓄積されている鍵データekによる最終鍵依存変換部308で変換し、出力部309から暗号文として出力データＣを出力する。

復号については、暗号化処理手順と逆の手順をたどることによって、暗号文Ｃから平文Ｍが得られる。特に、最終鍵依存変換部308が、初期鍵依存変換部302の逆変換になっているならば、図４において入力データの代りに暗号文データを入力し、鍵データを図４とは逆に、ek, $k_{(N-1)0}$, $k_{(N-1)1}$, $k_{(N-1)2}$, …, k_{10}, k_{11}, k_{12}, k_{00}, k_{01}, k_{02}, fkを順次与えればよい。

次に、非線形関数部304の内部を詳細に説明する。図５は、非線形関数部304の内部の機能構成を抜き出して示したものである。

第ｉ段ラウンド処理部38_iの入力ブロックデータR_iは、鍵記憶部322に蓄積されている鍵データk_{i0}, k_{i1}, k_{i2}とともに非線形関数部304への入力データとなる。ブロックデータR_iは、鍵データk_{i0}による第１鍵依存線形変換部341により例えば排他的論理和がとられてデータ$R_i^* = R_i \oplus k_{i0}$に線形変換される。次に、この変換されたデータ$R_i^*$は分割部342において例えば８ビットづつの４つのデータin_0, in_1, in_2, in_3にビット分割される。４つのデータin_0, in_1, in_2, in_3は、それぞれ非線形変換部343_0, 343_1, 343_2, 343_3において、データmid_{00}, mid_{01}, mid_{02}, mid_{03}に非線形変換された後、第２鍵依存線形変換部344に入力される。

第２鍵依存線形変換部344では、４つのデータ系統から入力されたデータmid_{00}, mid_{01}, mid_{02}, mid_{03}を系統間で互いに線形処理（排他的論理和）して新たな４つの系統のデータとし、更にそれらの系列のデータを鍵データk_{i1}の４つの部分により

それぞれ線形処理（排他的論理和）して４つの系統のデータmid_{10}, mid_{11}, mid_{12}, mid_{13}を出力する。これら４つのデータは非線形変換部345_0, 345_1, 345_2, 345_3に入力され、それぞれデータout_0, out_1, out_2, out_3とされる。これら４つの出力データは結合部346 で結合されてデータY_i^* とされ、更に第３鍵依存線形変換部347 においてデータY_i^* と鍵データk_{i2} との線形処理によりデータY_iを生成して出力する。

上述の第２鍵依存線形変換部344 は、データmid_{00}, mid_{01}, mid_{02}, mid_{03}に対応して設けられたデータ処理系統30_0, 30_1, 30_2, 30_3 間でこの発明によるアルゴリズムに従ってデータの排他的論理和をとることにより、図４のデータ変換装置のラウンド数を増やさずに安全性を高めるよう構成される。図４のデータ拡散装置の差分解読及び線形解読に対する安全性は、各ラウンドの非線形関数部304 の構成に依存し、特に図５に示す非線形関数部301 のが図６に示すような基本構成のとき、ｍビットの入力データがそれぞれ与えられるｎ個の非線形変換部（S-box）から成る第１非線形変換部343 と、それらのｎ個の出力を線形変換する線形変換部344Aと、そのｎ個のｍビット出力をそれぞれ非線形変換するｎ個の非線形変換部（S-box）から成る第２非線形変換部345 とによって安全性が決まる。特に、いかにして差分解読及び線形解読に対する耐性が強い最適な線形変換部344Aを構成するかが重要である。この発明によれば、線形変換部344Aを{0, 1}上のｎ×ｎ行列Ｐとして表現し、その行列Ｐの要素を最大差分確率・線形確率ｐ、ｑが最小となるように決定していくことにより、最適な線形変換部344Aを構成する。ただし、第２鍵依存線形変換部344 に含まれる副鍵k_{i1} による変換部は行列Ｐにより決定した線形変換部344Aに対し図７に示すように鍵依存変換部344Bとして付加することとする。

なお、”最適である”とは、上記の構造を有する線形変換部344Aの中で差分解読・線形解読に対する耐性が最も強くなるということであり、必ずしも他の指標、例えばアバランシュ性などについても最適であることを意味するわけではない。しかし、経験的にいえば、一般に段数（ラウンド数）を増やすだけでも差分解読・線形解読以外の攻撃を容易に回避できる一方、差分解読・線形解読については注意深くラウンド関数の特性を検討しなければ段数を多少増やしただけで攻撃を回避できるかどうかはわからない。そこで、この発明では、ラウンド関数の差分解

読・線形解読に対する耐性を最も重視し、最適な線形変換部344Aを求める。

この発明によれば、図６において上述のように線形変換部344Aを{0, 1}上のｎ×ｎ行列Ｐとして表現する。この行列Ｐは、ｍビット単位での線形変換であり、その線形変換部344Aが排他的論理和XOR だけで構成できることを意味する。即ち、この変換を次式

$$z'_i = \bigoplus_{j=0}^{n-1} t_{ij} z_j \qquad (1)$$

で表すことができる。特に、ｍ＝８の場合はバイト単位での線形変換となり、ワード幅が８ビット以上のいずれのプラットフォームにおいても効率的な実装が可能となる。

例えばｎ＝４の場合の具体例として、次式

$$\begin{bmatrix} z'_0 \\ z'_1 \\ z'_2 \\ z'_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} \qquad (2)$$

で表される４×４行列P_Eを説明する。行列P_Eを用いて構成したラウンド関数は以下の性質を有する。ただしS-box は全単射であると仮定する。上記行列式により規定されるz'_0, z'_1, z'_2, z'_3はそれぞれ以下の演算を表している。

$$z'_0 = 0\cdot z_0 \oplus 1\cdot z_1 \oplus 1\cdot z_2 \oplus 1\cdot z_3 = z_1 \oplus z_2 \oplus z_3 \qquad (3\text{-}1)$$

$$z'_1 = 1\cdot z_0 \oplus 0\cdot z_1 \oplus 1\cdot z_2 \oplus 1\cdot z_3 = z_0 \oplus z_2 \oplus z_3 \qquad (3\text{-}2)$$

$$z'_2 = 1\cdot z_0 \oplus 1\cdot z_1 \oplus 1\cdot z_2 \oplus 0\cdot z_3 = z_0 \oplus z_1 \oplus z_2 \qquad (3\text{-}3)$$

$$z'_3 = 1\cdot z_0 \oplus 1\cdot z_1 \oplus 1\cdot z_2 \oplus 1\cdot z_3 = z_0 \oplus z_1 \oplus z_2 \oplus z_3 \qquad (3\text{-}4)$$

差分解読及び線形解読に対するラウンド関数の耐性は、active s-box の最少個数n_d, n_l から決めることができ、これらの値は行列Ｐを決定した時点で決まる値である（付録参照）。差分解読法では入力差分値Δxが非零であるs-box をactive s-box と呼び、線形解読法では出力マスク値Γyが非零であるs-box をactive s-box と呼ぶ。

一般に、ある行列Ｐが与えられたとき、それに対応する線形変換部344Aの構成

は複数存在する。なぜなら、行列Ｐは線形変換部344Aの入力データと出力データとの関係を表しているだけであり、線形変換部344Aの具体的構成を規定しているわけではない。線形変換部344Aの構成が異なっても、線形変換部344Aの入力データと出力データとの関係を表す行列Ｐが同じであるならば、それらは同じ特性を持つものと判断できる。従って、以下では差分解読及び線形解読に対する耐性が高く、かつアバランシュ性がよくなるような行列Ｐを決めた後に線形変換部344Aの構成を決めることにする。その方が、線形変換部344Aの個々の構成について最適な特性を有するかどうかを調べて選択していくより効率よく、最適な特性を有する線形変換部344Aを見つけることができるためである。

ｎ×ｎ行列Ｐの要素は、差分特性に注目して以下の探索アルゴリズムによって決定される。

Step 1：安全性閾値Ｔ（Tは$2 \leqq T \leqq n$なる整数）を設定する。

Step 2：ハミング重みがT-1 以上となる列ベクトルの集合Ｃを用意する。具体的には要素"1"の数がT-1 以上のｎ次元列ベクトルをｎ個以上用意する。

Step 3：集合Ｃからｎ個の列ベクトルの組P_cを選択する。全ての組の検査を完了するまで以下の処理を繰り返す。

Step 3-1：列ベクトルの組P_cについて、n_dを求める。このことを$n_d(P_c)$と表す。

Step 3-2：$n_d(P_c) \geqq T$ ならば、そのｎ個の列ベクトルで構成される行列P_cを候補行列として採用する。

Step 4：全ての候補行列の中で、最大のn_dの値を与えた行列Ｐと$n_d(P)$の値を出力する。

上記探索アルゴリズムによる候補行列を採用するならば、n_dがＴ以上の値を持つことが保証される。n_dが最大となるような行列Ｐを求めるには上記探索アルゴリズムを実行する毎にＴをT=n, n-1, …, 3, 2 の順に１ずつ減らしていくことで、効率よく行列Ｐを発見できる。

上記探索アルゴリズムにおいて、差分解読及び線形解読に対しある程度満足いく耐性が得られるのであれば、Step 3-2まで実行して$n_d(P_c) \geqq T$となる行列を目的

とする行列Ｐとして使用してもよい。あるいは、Step 1を実行し、Step 2でハミング重みがT-1以上となるベクトルをｎ個選択し、それにより構成した行列Pcを目的とする行列Ｐとして使用してもよい。

線形変換部344Aの入力マスク値は、線形変換部344Aの出力マスク値の排他的論理和によって表現可能であるので、差分特性の場合と同様に、ある行列として表現できる。幾つかの線形変換部344Aの構成について、差分特性に注目した行列と線形表現に注目した行列との関係を調べた結果、以下の２つの予測ができた。

予測１：線形変換部344Aが{0, 1}上のｎ×ｎ行列Ｐで与えられたとする。このとき、線形変換部344Aの入力差分値Δｚと出力差分値Δｚ'との関係（差分値パス）は行列Ｐで与えられ、入力マスク値Γｚと出力マスク値Γｚ'との関係（マスク値パス）は転置行列^{T}Pで与えられる。即ち、

$$\Delta z' = P\Delta z \qquad (4)$$

$$\Gamma z = {}^{T}P\Gamma z' \qquad (5)$$

と表すことができる。

予測２：行列Ｐを用いた差分値パスにおけるactive s-boxの最少個数n_dと行列Ｐの転置行列^{T}Pを用いたマスク値パスにおけるactive s-boxの最少個数n_lは同じである。

上記予想２から、前述の探索アルゴリズムによる候補行列を採用するならば、n_lもＴ以上の値を持つ。例えば先の行列P_Eの場合、差分値パスに対する行列P_Eと、マスク値パスに対する行列${}^{T}P_E$は以下のようになる。

$$P_E = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \Leftrightarrow {}^{T}P_E = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \qquad (6)$$

この２つの行列について、それぞれn_d=3, n_l=3であることを証明することができる（付録参照）。

行列Ｐが与えられたとき、それに対応する線形変換部344Aの構成を決めるアルゴリズムを以下に示す。ここでは、

(1) 排他的論理和(XOR) の個数を最少にする、又は

(2) 同じような副構成が繰り返し現れる、

を満たすようにする。

Step 1：行列Pにおいて、２つの行を選択し、一方の行（a行）にもう一方の行（b行）を加える（基本演算と呼ぶ）。

Step 2：基本演算のみを用いて、行列Pが単位行列Iになるまで行を選択し、基本演算を行うことを繰り返し、そのときに行った基本演算の回数を数え、基本演算の回数が最少となるような行列変形手順を求める。

Step 3：線形変換部344Aを構成するために、Step 2で求めた変形手順の逆順に、そのときの基本演算で選択したa行、b行に相当するAライン、BラインとのXOR結線を行う。

この様にして決められた線形変換部344Aを有する第２鍵依存形線形変換部344の具体例を例えば図７に示す。線形変換部344Aにおいて、データmid_{00}, mid_{01}, mid_{02}, mid_{03} はそれぞれ処理系30_0 ～30_3 に入力され、処理系30_0 でmid_{00} とmid_{01} との排他的論理和が XOR回路31_0 でとられ、また処理系30_2 でmid_{02} とXOR回路31_0 の出力との排他的論理和が XOR回路31_2 でとられ、更にその回路31_2 の出力は処理系30_1 のXOR回路31_1でmid_{01} との排他的論理和がとられる。

処理系30_3 でXOR回路31_0 の出力とmid_{03} との排他的論理和が XOR回路31_3 でとられ、処理系30_1 でXOR回路31_1 の出力とXOR回路31_3 の出力との排他的論理和が XOR回路32_1 でとられ、処理系30_0 でXOR回路32_1 の出力とXOR回路31_0 の出力との排他的論理和が XOR回路32_0 でとられる。

XOR回路32_0, 32_1, 31_2, 31_3 の各出力と副鍵データk_{i10}, k_{i11}, k_{i12}, k_{i13} との各排他的論理和が 鍵依存変換部344BのXOR回路35_0～35_3 でそれぞれとられて、それぞれmid_{10}, mid_{11}, mid_{12}, mid_{13}を出力する。つまりデータmid_{00}, mid_{01}, mid_{02}, mid_{03} は相互に関連づけられた後、それぞれ各８ビットの副鍵データk_{i10}, k_{i11}, k_{i12}, k_{i13} に依存した線形変換が行われる。論理式で示すと下記の論理演算がなされる。

$$mid_{10}=mid_{00}\oplus mid_{02}\oplus mid_{03}\oplus k_{i10} \qquad (7\text{-}1)$$

$$mid_{11}=mid_{01}\oplus mid_{02}\oplus mid_{03}\oplus k_{i11} \qquad (7\text{-}2)$$

$$mid_{12}=mid_{00}\oplus mid_{01}\oplus mid_{02}\oplus k_{i12} \quad (7-3)$$

$$mid_{13}=mid_{00}\oplus mid_{01}\oplus mid_{03}\oplus k_{i13} \quad (7-4)$$

なお、副鍵k_{i1}は４つのデータk_{i10}，k_{i11}，k_{i12}，k_{i13}で構成されている。

図５に示したように、次に、これらデータmid_{10}，mid_{11}，mid_{12}，mid_{13}は、それぞれ非線形変換部345_0，345_1，345_2，345_3において、データout_0，out_1，out_2，out_3に非線形変換された後、結合部346 において、一つのデータY_i^* に結合される。つまり例えば４つの８ビットのデータが１つの３２ビットデータにビット結合される。最後に、データY_i^* は、鍵データk_{i2}による第３鍵依存線形変換部347 において、例えばデータY_i^* をk_{i2}ビット左ローテーションによりデータY_iに線形変換され、非線形関数部304 からの出力データY_iが生成される。非線形変換部343_0～343_3，345_0 ～345_3 のそれぞれは、例えばＤＥＳ暗号のS-box のようなものであり、例えばＲＯＭにより構成され、それぞれ入力データをアドレスとして入力し、対応するデータを読み出すものである。

ここで、非線形変換部343_0～343_3は４つ並列に配置されており、その変換処理は相互に関連していないため、これらは並列実行が可能である。また、非線形変換部345_0～345_3についても同様のことがいえる。このため、これら両非線形変換はそれぞれ１ステップ（非線形関数部304 としては合計２ステップ）で実行することができる。また、非線形変換部343_0～343_3，345_0～345_3の差分確率・線形確率がｐとすると、第２鍵依存線形変換部344 に図７に示したものを用いた場合、非線形関数部304 全体における差分確率・線形確率はp^4 となる。従って、データ変換装置全体では段数（ラウンド数）を3rとして確率$P\leqq p^{8r}$で近似表現でき、例えばｒ＝４（段数１２段）とすると$P\leqq p^{32}$である。これはＤＥＳ暗号に換算すると４８段以上に相当し、差分解読法および線形解読法に対して十分安全なデータ変換装置となる。

なお、鍵データfk，k_{00}，k_{01}，k_{02}，k_{10}，k_{11}，k_{12}，…，$k_{(N-1)0}$，$k_{(N-1)1}$，$k_{(N-1)2}$，ek は図４において鍵生成スケジュール部２０の鍵入力部320 から入力された主鍵情報Ｋey から鍵データ生成部321 によって変換され、鍵記憶部322 に蓄積されたデータである。鍵データ生成部321 による鍵データの生成は、図１に示したＤＥＳ暗号

の拡大鍵生成ルーチン部21と同様に構成してもよいし、あるいは図３に示した宮口らの拡大鍵生成ルーチン部21と同様に構成してもよい。

図４に示した初期鍵依存変換部302 、最終鍵依存変換部308 、及び図５に示した各非線形関数部304 内の鍵依存線形変換部341, 344, 347 は鍵に依存する線形変換部であるため、差分解読法および線形解読法以外の解読法に対しても十分な安全性を兼ね備え、最も安全性を重視した暗号装置である。

なお、この発明はこの例に特定されるだけでなく、例えば高速性を望むのであれば、これら初期鍵依存変換部302 、最終鍵依存変換部308 、鍵依存線形変換部341, 344, 347 については、そのいずれか１つを削除する、又は鍵に依存しない変換手段に変更することが可能である。この場合、差分解読法および線形解読法に対する安全性をそれほど低下させることなく暗号化処理速度の向上が望める。

第２実施例

図４に示した第１実施例と同様の構成を有するデータ変換装置において、図５に示す非線形関数部304 として他の機能構成を用いた別の実施例を説明する。この実施例は、図５における非線形変換部343_0, 343_1, 343_2, 343_3の代わりに、それぞれ例えば８ビットの入力in_0～in_3に対して３２ビットの拡大データMID_{00}, MID_{01}, MID_{02}, MID_{03}を非線形変換により出力する等価的に図８Ａ～８Ｄで示す非線形変換部$343_0'$～$343_3'$を使用し、鍵依存線形変換部344 として図９に示すものを使用する。

図５に示したのと同様にデータR_iは、鍵記憶部322 に蓄積されている鍵データk_{i0}, k_{i1}, k_{i2}と共に非線形関数部304 への入力データとなる。データR_iは、鍵データk_{i0}による第１鍵依存線形変換部341 で、例えば排他的論理和によりデータ$R_i^* = R_i \oplus k_{i0}$に線形変換される。次に、データ$R_i^*$は分割部342 において４つのデータ$in_0$, in_1, in_2, in_3に分割される。４つのデータin_0, in_1, in_2, in_3は、それぞれ図８Ａ～８Ｄに示す非線形変換部$343_0'$, $343_1'$, $343_2'$, $343_3'$において、データMID_{00}, MID_{01}, MID_{02}, MID_{03}に非線形変換される。第１実施例では、非線形変換部343_0がｍビット入力in_0に対し、ｍビットのデータmid_{00}を出力したのに対し、ここでは、非線形変換部$343_0'$は、第１実施例の図５における非線形変換部343_0と同じｍビットの

データmid_{00} を上位mビットとして出力すると共に、下位mビットに固定データ"0 $0\cdots0_{(2)}$"を出力するよう構成されたS-box を有し、その上位mビットデータmid_{00} を布線により３系統に出力すると共にmビットの"$00\cdots0_{(2)}$"を出力するよう構成されている。即ち、非線形変換部$343_0'$はmビットのデータin_0 を4mビットのデータ

$$MID_{00}=[mid_{00},\ 00\cdots0_{(2)},\ mid_{00},\ mid_{00}] \qquad (8\text{-}1)$$

に変換する手段である。同様に、非線形変換部$343_1'$, $343_2'$, $343_3'$ はそれぞれin_1, in_2, in_3 から

$$MID_{01}=[00\cdots0_{(2)},\ mid_{01},\ mid_{01},\ mid_{01}] \qquad (8\text{-}2)$$

$$MID_{02}=[mid_{02},\ mid_{02},\ mid_{02},\ 00\cdots0_{(2)}] \qquad (8\text{-}3)$$

$$MID_{03}=[mid_{03},\ mid_{03},\ 00\cdots0_{(2)},\ mid_{03}] \qquad (8\text{-}4)$$

に変換する手段である。式(8-1)で表されるデータMID_{00} の決め方は、予め図７の線形変換部344Aにおいてデータmid_{00} 以外のデータmid_{01}, mid_{02}, mid_{03} 全てをそれぞれ"$00\cdots0_{(2)}$"とおいた場合に線形変換部344Aの出力４系統に得られるデータ全体をMID_{00} とすればよい。以下同様にして式(8-1), (8-2), (8-3) で表されるデータMID_{01}, MID_{02}, MID_{03} についてもそれぞれ容易に決めることができる。これら非線形変換部$343_0'$～$343_3'$はそれぞれデータin_0, in_1, in_2, in_3 をアドレスとし、データMID_{00}, MID_{01}, MID_{02}, MID_{03} を直接読み出す変換テーブルとしてメモリにより構成してもよい。

次いで、これらデータMID_{00}～MID_{03}は図９に示す鍵データk_{i1}による第２鍵依存線形変換部344 に入力される。MID_{00}とMID_{01}はXOR回路41で排他的論理和がとられ、MID_{02}とMID_{03}は XOR回路42で排他的論理和がとられ、XOR回路41, 42の両出力の排他的論理和がXOR回路４３でとられ、XOR回路４３の出力と鍵データk_{i1}との排他的論理和がXOR回路４４でとられる。XOR回路４４の出力MID_1は各mビットの出力mid_{10}, mid_{11}, mid_{12}, mid_{13}にビット分割されて出力される。結局、この第２鍵依存線形変換部344 は次式

$$MID_1=MID_{00}\oplus MID_{01}\oplus MID_{02}\oplus MID_{03}\oplus k_{i1} \qquad (9)$$

の演算で入力を線形変換する。

この線形変換演算による出力$MID_1=[mid_{10},\ mid_{11},\ mid_{12},\ mid_{13}]$の成分はそれぞ

れ次式で表される：

$$mid_{10}=mid_{00}\oplus mid_{02}\oplus mid_{03}\oplus k_{i10} \quad (10\text{-}1)$$

$$mid_{11}=mid_{01}\oplus mid_{02}\oplus mid_{03}\oplus k_{i11} \quad (10\text{-}2)$$

$$mid_{12}=mid_{00}\oplus mid_{01}\oplus mid_{02}\oplus k_{i12} \quad (10\text{-}3)$$

$$mid_{13}=mid_{00}\oplus mid_{01}\oplus mid_{03}\oplus k_{i13} \quad (10\text{-}4)$$

これは、図７における線形変換式(7-1)～(7-4)と等価な線形変換である。このようにして第１実施例と同じデータmid_{10}, mid_{11}, mid_{12}, mid_{13}が生成される。なお、k_{i1}は４つのデータk_{i10}, k_{i11}, k_{i12}, k_{i13}で構成されている。

次いで、データmid_{10}, mid_{11}, mid_{12}, mid_{13}は、図５に示したのと同様にそれぞれ非線形変換部345_0, 345_1, 345_2, 345_3において、データout_0, out_1, out_2, out_3に非線形変換された後、結合部346 において、４つのデータout_0, out_1, out_2, out_3が１つのデータY_i^*に結合される。最後に、データY_i^*は、鍵データk_{i2}による第３鍵依存線形変換部347 において、例えばk_{i2}ビット左ローテーションによりデータY_iに線形変換され、非線形関数部304 からの出力データY_iが生成される。

図８Ａ～８Ｄ、９に示した第２実施例において、図８Ａ～８Ｄの非線形変換部343_0～343_3は第１実施例と同様にそれぞれ各８ビットのデータmid_{00}～mid_{03}を出力するS-box のみで構成し、図８Ａ～８Ｄで示した布線と８ビットの"00…0"を出力するレジスタを図９の鍵依存線形変換部344 内に設け、その中でデータMID_{00}～MID_{03}を生成するようにしてもよい。

この第２実施例における第２鍵依存線形変換部344 では、図９に示したように４つ排他的論理和により図７と等価な線形変換（図７では排他的論理和の個数は１０個）を実現しているため、第１実施例より高速な変換が可能になる。

また、第１実施例と同様に、非線形変換部343_0～343_3と345_0～345_3は４つずつ並列に配置されており、その非線形変換処理は相互に関連していないため、これらはすべて並列実行が可能である。更に、非線形変換部343_0～343_3, 345_0～345_3の差分確率・線形確率がｐとすると、非線形関数部304 全体における差分確率・線形確率はp^4となる。

第３実施例

第１実施例と同様に、図４に示す機能構成を有するデータ変換装置において、その非線形関数部304 の内部の機能構成として更に別のものを用いた実施例を説明する。

図５に示したように例えば３２ビットのデータR_iは、鍵記憶部322 に蓄積されている鍵データk_{i0}, k_{i1}, k_{i2} とともに非線形関数部304 へ入力される。データR_iは、鍵データk_{i0} による第１鍵依存線形変換部341 で、例えば排他的論理和によりデータ$R_i^* = R_i \oplus k_{i0}$ に線形変換される。次に、データR_i^* は分割部342 において例えば８ビットの４つのデータin_0, in_1, in_2, in_3 に分割される。

非線形変換部343_0 において、例えば図１０に示すように、in_0 は更に例えば４ビットの２つのサブブロックin_{00} とin_{01} に分割された後、in_{00} はサブ非線形変換部５１によりデータmid_{000} に変換され、またin_{01}との排他的論理和がXOR回路５２でとられ、その出力$in_{00} \oplus in_{01}$ はサブ非線形変換部５３によりデータmid_{001}に変換される。ついで、これら出力mid_{000}とmid_{001}の排他的論理和がXOR回路５４でとられ、そのXOR回路５４の出力とmid_{001}を統合してmid_{00} に変換する。つまり、非線形変換部343_0は入力in_0 を２つのサブブロックに分割し、これら２つのサブブロックについて、それぞれ線形変換と非線形変換を行い、その２つの出力サブブロックを統合して非線形変換部の出力とする。同様に、in_1 in_2 in_3についても、それぞれ２つの非線形変換部と２つの排他的論理和回路とよりなる図１０に示した機能構成の非線形変換部343_1, 343_2, 343_3 を用いて、データmid_{01}, mid_{02}, mid_{03} に変換する。

変換されたデータmid_{00}, mid_{01}, mid_{02}, mid_{03}は、図７に示した鍵データk_{i1} による第２鍵依存線形変換部344 に入力される。この変換部344 によりそれぞれ前述の式(7-1)～(7-4)の演算が行われる。

次いで、データmid_{10}は、非線形変換部345_0 においても、例えば図１０に示したのと同様の機能構成により、更に２つのサブブロックmid_{100}とmid_{101} に分割され、mid_{100} はサブ非線形変換部５１によりデータout_{00} に変換され、一方、データmid_{100}とmid_{101}はXOR回路５２により排他的論理和がとられ、その出力$mid_{100} \oplus mid_{101}$ は非線形変換部５３によりデータout_{01} に変換される。ついで、データout_{00}とout_{01}の排他的論理和がXOR回路５４によりとられ、その出力$out_{00} \oplus out_{01}$ とout_{01} を統合してout_0

に変換する。同様に、mid_{11}, mid_{12}, mid_{13} についても、それぞれ２つのサブ非線形変換部５１，５３と２つの排他的論理和回路５２，５４とよりなる図１０に示したのと同様の機能構成の非線形変換部345_1, 345_2, 345_3 によりデータout_1, out_2, out_3 に変換される。

これら非線形変換された４つのデータout_0, out_1, out_2, out_3 は、結合部346 において一つのデータY_i^* に結合される。最後に、データY_i^* は、鍵データk_{i2} による第３鍵依存線形変換部347 において、例えばk_{i2} ビット左ローテーションによりデータY_iに線形変換され、非線形関数部304 からの出力データY_iが生成される。

この様に、この第３実施例においては、非線形変換部343_0～343_3, 345_0～345_3では、それぞれその内部で入力データが２つのデータに分割されて２つのサブ非線形変換部（図１０では５１，５３）により非線形変換される。このため、これら１６個の各非線形変換部がそれぞれ扱うことのできる２倍のビット長を、非線形変換部343_0～343_3, 345_0～345_3 への入力データとすることができる。例えば、サブ非線形変換部５１，５３を８ビット長のS-box とすると、非線形変換部343_0～343_3, 345_0～345_3 への各入力データは１６ビット長となり、非線形変換関数304 への入力データは６４ビット長となる。これにより、図４の構成を有するデータ変換装置として、ブロック長を128 ビットとすることができる。

また、サブ非線形変換部５１，５３対応のものは８つずつ並列に配置されており、その非線形変換処理は相互に関連していないため、これらはすべて並列実行が可能である。更に、サブ非線形変換部５１，５３の差分確率・線形確率がｐとすると、非線形関数部304 全体における差分確率・線形確率はp^4 となる。

上述において第１鍵依存線形変換部341，第２鍵依存線形変換部344 及び第３鍵依存線形変換部347 は必ずしも鍵依存とすることなく、つまり鍵データを入力することなく、サブデータ内での線形変換を行ってもよい。

前述の各実施例ではハードウェア構成として説明したが、そのデータ処理をプログラムに従って実行するソフトウェアにより実現してもよい。例えば、その処理手順の要部をフローチャートとして図１１に示す。図１１は図４の処理全体に対応する処理手順を表している。

ステップＳ１：処理繰り返し回数を表す変数ｉを０に初期化する。

ステップＳ２：入力平文を初期変換してから左右ブロックデータL_i，R_iに分割する。

ステップＳ３：副鍵k_iを使って非線形関数により右ブロックデータR_iを処理してブロックデータY_iを生成する。

ステップＳ４：左ブロックデータL_iをブロックデータY_iで線形処理してブロックデータL'_iを生成する。

ステップＳ５：右ブロックデータR_iを新しい左ブロックデータL_iとし、ブロックデータL'_iを新しい右ブロックデータR_iとする。

ステップＳ６：ｉを１だけインクリメントする。

ステップＳ７：ｉがＮに達したか判定し、達してなければステップＳ３に戻り、前述と同様の処理ステップＳ３～Ｓ７を繰り返す。

ステップＳ８：前ステップＳ７でｉがＮに達していると判定されると、左右データL_i，R_iを結合し、更に最終変換した結果を拡散データＣとして出力する。

図１１のステップＳ３の処理の詳細は図５の非線形関数部304による処理に対応し、その処理手順を図１２に示す。

ステップＳ３１：右データR_iに対し第１の鍵依存線形変換を行い、データR'_iとする。

ステップＳ３２：データR'_iをｍビットずつｎ分割してデータin_0，in_1，…，in_{n-1}を生成する（例えばm=8，n=4である）。

ステップＳ３３：データin_0，in_1，…，in_{n-1}をそれぞれアドレスとしてｎ個の第１のS-boxからそれぞれデータmid_{00}，mid_{01}，…，$mid_{0(n-1)}$を読み出す。

ステップＳ３４：データmid_{00}～$mid_{0(n-1)}$に対し副鍵k_{i1}を使った鍵依存線形変換を行い、データmid_{10}～$mid_{1(n-1)}$を生成する。

ステップＳ３５：データmid_{10}～$mid_{1(n-1)}$をそれぞれアドレスとしてｎ個の第２のS-boxからデータout_0～out_{n-1}を読み出す。

ステップＳ３６：データout_0～out_{n-1}を結合してデータY'_iを得る。

ステップＳ３７：データY'_iに対し、第３の鍵依存線形変換を行ってデータY_iを

生成し、出力する。

上記ステップＳ３４の演算は、式(7-1)～(7-4)による演算でもよいし、又は式(8-1)～(8-4)の定義を使った式(9) による演算でもよい。更に、図１１ではステップＳ３乃至Ｓ７をラウンド数だけ繰り返し実行する処理手順の場合を示したが、図３に示したＮ段のラウンド処理部38_0～38_{N-1}のそれぞれの処理をそのままプログラム化してこの発明のデータ拡散を実現してもよい。

第４実施例

前述の図４に示した第１実施例は、その非線形関数部304 （図５）の第２鍵依存線形変換部344 を構成する図６で説明した基本線形変換部344Aを４×４の行列で表す場合（即ち、４入力、４出力の場合）の実施例であったが、以下に示す第４実施例では、線形変換部344Aを８×８行列で表す場合について説明する。

図１３は、この発明の第４実施例を示すデータ変換装置における、暗号化処理手順の機能構成を示したものである。この構成自体は図４に示した第１実施例のものと同じであるが、異なる点は、データのビット数と、非線形関数部304 におけるデータの分割数ｎである。

入力データＭは、鍵記憶部322 に蓄積されている鍵データfkによる初期鍵依存変換部302 で変換された後、初期分割部303 で左、右ブロックデータL_0, R_0 に分割される。例えば128 ビットのデータが６４ビットずつのブロックデータL_0, R_0 に分割される。初期鍵依存変換部302 では、例えば鍵データfkと入力データＭとの排他的論理和や鍵データfkによる入力データＭのビットローテーション（ビット回転）などの線形変換、もしくは乗算などを組み合わせた非線形変換などが行われる。

右ブロックデータR_0は、鍵記憶部322 に蓄積されている鍵データk_{00}, k_{01}, k_{02} とともに非線形関数部304 に入力され、非線形関数部304 で非線形変換処理が行われてデータY_0に変換される。データY_0とデータL_0は線形演算部305 で線形演算されてデータL_0^* に変換される。データL_0^* とデータR_0は交換部306 でデータ位置の交換が行われ、$L_1 \leftarrow R_0$, $R_1 \leftarrow L_0^*$ とされ、L_1, R_1が次の第１段ラウンド処理部381 に入力

される。

以下、第ｉ段ラウンド処理部38_i (i=0, 1, …, N-1)において、左右ブロックデータL_i, R_iについて上記と同様の処理を繰り返し行う。即ち、左右ブロックデータL_i, R_i について、右ブロックデータR_iは、鍵記憶部322 に蓄積されている鍵データk_{i0}, k_{i1}, k_{i2} と共に非線形関数部304 に入力され、非線形関数部304 で非線形変換処理が行われて、ブロックデータY_iに変換される。ブロックデータY_iとL_iは線形演算部305 で演算されてデータL_i^* に変換される。データL_i^* とデータR_iは交換部306 でデータ位置の交換が行われ、$L_{i+1} \leftarrow R_i$, $R_{i+1} \leftarrow L_i^*$ のように交換される。線形演算部305 は例えば排他的論理和演算を行うものである。

データ変換装置としての安全性を確保するための適切なラウンド繰り返し回数をＮとすると、繰り返し処理の結果、左右ブロックデータL_N, R_N が得られる。これらのブロックデータL_N, R_N を最終結合部307 で結合し、つまり例えば各６４ビットのブロックデータL_N, R_N を結合して128 ビットのデータとし、その後、鍵記憶部322 に蓄積されている鍵データekによる最終鍵依存変換部308 で変換し、出力部309 から暗号文として拡散データＣを出力する。

復号については、暗号化処理手順と逆の手順をたどることによって、暗号文Ｃから平文Ｍが得られる。特に、最終鍵依存変換部308 が初期鍵依存変換部302 の逆変換になっているならば、図１３において入力データの代りに暗号文データを入力し、鍵データを図１３とは逆に、ek, $k_{(N-1)0}$, $k_{(N-1)1}$, $k_{(N-1)2}$, …, k_{10}, k_{11}, k_{12}, k_{00}, k_{01}, k_{02}, fkを順次与えればよい。

次に、非線形関数部304 の内部を詳細に説明する。図１４は、非線形関数部304 の内部の機能構成を抜き出して示したものである。

右ブロックデータR_i は、鍵記憶部322 に蓄積されている鍵データk_{i0}, k_{i1}, k_{i2} と共に非線形関数部304 への入力データとなる。右ブロックデータR_iは、副鍵データk_{i0} による第１鍵依存線形変換部341 、例えば排他的論理和によりデータ$R_i^* = R_i \oplus k_{i0}$ に線形変換される。この変換されたデータR_i^* は分割部342 において例えばｎ＝８個のデータin_0, in_1, in_2, …, in_7 に分割される。８つのデータin_0 ～in_7 は、それぞれ非線形変換部343_0 ～343_7 において、データmid_{00} ～mid_{07} に非線形変換さ

れた後、鍵データk_{i1}による第２鍵依存線形変換部344 に入力される。

第２鍵依存線形変換部344 では、８つのデータ系統から入力されたデータmid_{00}, mid_{01}, mid_{02}, …, mid_{07}を系統間で互いに線形処理（排他的論理和）して新たな８つの系統のデータとし、更にそれらの系列のデータを鍵データk_{i1} の８つの部分によりそれぞれ線形処理（排他的論理和）して８つの系統のデータmid_{10}, mid_{11}, mid_{12}, …, mid_{17}を出力する。これら８つのデータは非線形変換部345_0, 345_1, 345_2, …, 345_7に入力され、それぞれデータout_0, out_1, out_2, …, out_7とされる。これら８つの出力データは結合部346 で結合されてデータY_i^* とされ、更に第３鍵依存線形変換部347 においてデータY_i^* と鍵データk_{i2} との線形処理によりデータY_iを生成して出力する。

第２鍵依存線形変換部344 は、図６で説明したと同様にｎ×ｎ行列で表される線形変換部344Aを含んでおり、この第４実施例ではｎ＝８である。ただし、線形変換部は全単射であると仮定する。即ち、rank(P)=8である。以下に、第１実施例で説明したと同様に、n_dが最大となるような８×８行列Ｐを求める。ここでは、安全閾値ＴをT=8, 7, ... と順次減らし、各値毎に以下のアルゴリズムを実行する。

Step 1：安全閾値Ｔ（Ｔは$2 \leqq T \leqq n$なる整数）を設定する。

Step 2：ハミング重みがT-1以上となる列ベクトルの集合Ｃを用意する。

Step 3：集合Ｃから８個の列ベクトルの組Pcを選択する。このとき、$rank(P_c) \neq 8$ならば、そのP_cは候補としない。

Step 3-1：P_cに対するn_dを以下のようにして求める。

・任意の２列（a, b列）について

$$n_{d0} = 2 + \min_{(a,b)} \#\{(t_{ia}, t_{ib}) \mid t_{ia} \oplus t_{ib} \neq 0, 0 \leqq i < 8\}$$

・任意の３列（a, b, c列）について

$$n_{d1} = 3 + \min_{(a,b,c)} \#\{(t_{ia}, t_{ib}, t_{ic}) \mid t_{ia} \oplus t_{ib} \oplus t_{ic} \neq 0, 0 \leqq i < 8\}$$

$$n_{d2} = 3 + \min_{(a,b,c)} \#\{(t_{ia}, t_{ib}, t_{ic}) \mid (0,0,0),(1,1,1) \text{を除く}, 0 \leqq i < 8\}$$

・任意の４列（a, b, c, d列）について

$$n_{d3} = 4 + \min_{(a,b,c,d)} \#\{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid \begin{matrix}(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0)\\(0,1,1,1),(1,0,1,1),(1,1,0,1),(1,1,1,0)\end{matrix}, 0 \leqq i < 8\}$$

$$n_{d4} = 4 + \min_{(a,b,c,d)} \#\{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0),(1,1,0,0),(0,1,1,1),(1,0,1,1) \text{を除く}, 0 \leqq i < 8\}$$

$$n_{d5} = 4 + \min_{(a,b,c,d)} \#\{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0),(1,0,1,0),(0,1,1,1),(1,1,0,1) \text{を除く}, 0 \leqq i < 8\}$$

$$n_{d6} = 4 + \min_{(a,b,c,d)} \#\{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0),(1,0,0,1),(0,1,1,1),(1,1,1,0) \text{を除く}, 0 \leqq i < 8\}$$

$$n_{d7} = 4 + \min_{(a,b,c,d)} \#\{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0),(0,1,1,0),(1,0,1,1),(1,1,0,1) \text{を除く}, 0 \leqq i < 8\}$$

$$n_{d8} = 4 + \min_{(a,b,c,d)} \#\{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0),(0,1,0,1),(1,0,1,1),(1,1,1,0) \text{を除く}, 0 \leqq i < 8\}$$

$$n_{d9} = 4 + \min_{(a,b,c,d)} \#\{(t_{ia}, t_{ib}, t_{ic}, t_{id}) \mid (0,0,0,0),(0,0,1,1),(1,1,0,1),(1,1,1,0) \text{を除く}, 0 \leqq i < 8\}$$

・$n_d = \min\{n_{di} \mid 0 \leqq i \leqq 9\}$

直感的にいえば、式n_{d0}～n_{d9}は、第１非線形変換部343のactive s-boxの個数（右辺第１項）が決められたとき、第２非線形変換部345で最小いくつ以上（右辺第２項）のs-boxがactiveになり、そのときのactive s-boxの合計がいくつ以上（左辺）になるのかを表している。例えば、第１非線形変換部343で２つのactive s-boxがある場合、その差分値をそれぞれΔz_a, Δz_bと表すことができる。このとき、

$$[\Delta z'_i] = [t_{ia}\Delta z_a \oplus t_{ib}\Delta z_b] \quad (0 \leqq i < 8) \qquad (11)$$

となる。特に、$\Delta z_a = \Delta z_b$とすると、

$$[\Delta z'_i] = [(t_{ia} \oplus t_{ib})\Delta z_a] \quad (0 \leqq i < 8) \qquad (12)$$

となる。従って、この場合のactive s-boxの最少個数はn_{d0}で与えられることになる。

上述の探索アルゴリズムにより、行列Pを探索した結果、$n_d \geqq 6 = T$となるような行列は存在せず、また、$n_d = 5 = T$となる行列は10080個存在した。その結果、これら

の行列Ｐを用いたラウンド関数の差分解読法に対する耐性は$p \leqq p_s^5$ となる。また、線形解読法に対する耐性も$q \leqq q_s^5$ となる。

上記の10080 個の行列Ｐについて、その構成を決める。全数探索的に構成を決めることは例えば１６個のXOR結線を使うとして、$(8\times7)^{16} \fallingdotseq 2^{93}$ 程度の計算量が必要であり、実行不可能である。そこで、図１５Ａに示すように、線形変換部344Aの内部が８入力４出力のボックスB1～B4で構成される構造に限定する。各ボックスの内部構成は図１５Ｂに示すように、４つのXOR回路で構成されており、全ての入力ラインが１回ずつXOR回路を通過するのもとする。従って、線形変換部344A全体では、１６個のXOR回路で構成されることになる。このとき、$(4\times3\times2\times1)^4 \fallingdotseq 2^{19}$程度の計算量となり、十分に全数探索が実行可能となる。

図１５Ａでは左右各４系統のラインに対し４つの変換ボックスB1～B4が交互に挿入されているが、これらのラインは任意に選択した４ラインと、その残りの４ラインに決めてよい。各変換ボックスにはそれが挿入されている４ラインからの入力と、残りの４ラインからの入力が与えられ、変換結果を前者の４ラインに出力する。

上述の探索アルゴリズムにより得た10080 個の行列のうち、図１５の構造を満たしつつ、１６回の基本演算（XOR)で単位行列Ｉとなるものがあるかどうかを探索した結果、５７個の構成が見つかった。その中の１つの行列Ｐを次に示す。

$$P=\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad (13)$$

この行列を使用した線形変換部344Aの構成例を非線形変換部343，345 と共に図１６に示す。図に示すように、第１非線形変換部343 を構成する８個のS-box からの８ラインの左右４系統のラインに対し、４つの変換ボックスB1～B4が交互に挿入されており、その結果、各ラインには２個のXOR回路が挿入されている。

第１実施例において４×４行列の場合に与えたと同様の予測、即ち図１６で構成された線形変換部344Aにおいて、マスク値パスに対する行列が行列Ｐの転置行列となっていること、及び$n_1=5$ となることが正しく成立するかを以下のように確認することができる。図１６で構成された線形変換部344Aに、付録に定理２で示すconcatenation rules を用いて線形パスを構成することによって、マスク値パスに対する行列${}^{T}P$は以下のように求められる。

$$ {}^{T}P = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \qquad (14) $$

このことから、行列${}^{T}P$が行列Ｐの転置行列となっていることがわかる。更に、active s-box の最少個数を調べると、$n_1=5$となることが確認できる。

図１７は、上述のようにして決定された線形変換部344Aと、更に鍵変換部344Bとを含む第２鍵依存線形変換部344 の具体例を示す。

鍵変換部344Bは鍵データk_{i10}, k_{i11}, k_{i12}, …, k_{i17}と線形変換部の出力との排他的論理和演算をXOR回路63_0, 63_1, 63_2, …, 63_7 で行い、出力データmid_{10}, mid_{11}, mid_{12}, …, mid_{17} を出力するように構成されている。例えば、図１７に示すような機能的構成により、次の演算がなされる。

$$mid_{10}=mid_{01}\oplus mid_{02}\oplus mid_{03}\oplus mid_{04}\oplus mid_{05}\oplus mid_{06}\oplus k_{i10} \qquad (15\text{-}1)$$

$$mid_{11}=mid_{00}\oplus mid_{02}\oplus mid_{03}\oplus mid_{05}\oplus mid_{06}\oplus mid_{07}\oplus k_{i11} \qquad (15\text{-}2)$$

$$mid_{12}=mid_{00}\oplus mid_{01}\oplus mid_{03}\oplus mid_{04}\oplus mid_{06}\oplus mid_{07}\oplus k_{i12} \qquad (15\text{-}3)$$

$$mid_{13}=mid_{00}\oplus mid_{01}\oplus mid_{02}\oplus mid_{04}\oplus mid_{05}\oplus mid_{07}\oplus k_{i13} \qquad (15\text{-}4)$$

$$mid_{14}=mid_{00}\oplus mid_{01}\oplus mid_{03}\oplus mid_{04}\oplus mid_{05}\oplus k_{i14} \qquad (15\text{-}5)$$

$$mid_{15}=mid_{00}\oplus mid_{01}\oplus mid_{02}\oplus mid_{05}\oplus mid_{06}\oplus k_{i15} \qquad (15\text{-}6)$$

$$mid_{16}=mid_{01}\oplus mid_{02}\oplus mid_{03}\oplus mid_{06}\oplus mid_{07}\oplus k_{i16} \qquad (15\text{-}7)$$

$$mid_{17}=mid_{00}\oplus mid_{02}\oplus mid_{03}\oplus mid_{04}\oplus mid_{07}\oplus k_{i17} \qquad (15\text{-}8)$$

この演算により、データmid_{10}, mid_{11}, mid_{12}, …, mid_{17}を生成する。なお、副鍵k_{i1}は８つのデータk_{i10}, k_{i11}, k_{i12}, …, k_{i17}で構成されている。図１７において、mid_{00}～mid_{07}はそれぞれ系路60_0～60_7に入力される。

系路60_4, 60_5, 60_6, 60_7上のXOR回路61_4, 61_5, 61_6, 61_7でそれぞれmid_{04}とmid_{00}、mid_{05}とmid_{01}、mid_{06}とmid_{02}、mid_{07}とmid_{03}との各排他的論理和算がなされる。

系路60_0, 60_1, 60_2, 60_3上のXOR回路61_0, 61_1, 61_2, 61_3でそれぞれmid_{00}とXOR回路61_6の出力、mid_{01}とXOR回路61_7の出力、mid_{02}とXOR回路61_4の出力、mid_{03}とXOR回路61_5との出力の各排他的論理和算がなされる。

系路60_4, 60_5, 60_6, 60_7上の回路62_4, 62_5, 62_6, 62_7で、それぞれXOR回路61_3と61_4の各出力、XOR回路61_0と61_5の各出力、XOR回路61_1と61_6の各出力、XOR回路61_2と61_7の各出力の排他的論理和算がなされる。

系路60_0, 60_1, 60_2, 60_3上のXOR回路62_0, 62_1, 62_2, 62_3で、それぞれXOR回路61_0と62_4の各出力、XOR回路61_1と62_5の各出力、XOR回路61_2と62_6の各出力、XOR回路61_3と62_7の各出力排他的論理和算がなされる。

更に、系路60_0～60_7上のXOR回路63_0～63_7で、XOR回路62_0～62_7の各出力と鍵データk_{i10}～k_{i17}との排他的論理和算がそれぞれとられ、系路60_0～60_7からそれぞれ出力mid_{10}～mid_{17}が出力される。つまり出力mid_{10}～mid_{13}は、入力mid_{00}～mid_{07}から選んだ６つと鍵データとの排他的論理和算がなされ、出力mid_{14}～mid_{17}は入力mid_{00}～mid_{07}から選んだ５つと鍵データとの排他的論理和算がなされる。

次いで、データmid_{10}, mid_{11}, mid_{12}, …, mid_{17}は、図１４でそれぞれ非線形変換部345_0, 345_1, 345_2, …, 345_7において、データout_0, out_1, out_2, …, out_7に非線形変換された後、結合部346 において、８つのデータout_0, out_1, out_2, …, out_7は１つのデータY_i'に結合される。最後に、データY_i'は、鍵データk_{i2}による第３鍵依存線形変換部347 において、例えばk_{i2}ビット左ローテーションによりデータY_iに線形変換され、非線形関数部304 からの出力データY_iが生成される。

非線形変換部343_0～343_7, 345_0～345_7のそれぞれは、例えばＤＥＳ暗号のS-boxのようなものであり、例えばＲＯＭにより構成され、それぞれ入力データをアドレスとして入力し、対応するデータを読み出すものである。

ここで、非線形変換部343_0～343_7は８つ並列に配置されており、その非線形変換処理は相互に関連していないため、これらはすべて並列実行が可能である。また、非線形変換部345_0～345_7についても同様のことがいえる。このため、非線形変換はそれぞれ１ステップ（合計２ステップ）で実行することができる。

また、非線形変換部343_0～343_7の差分確率・線形確率がｐとすると、第２鍵依存線形変換部344 に図１７に示した線形変換を用いた場合、非線形関数部304 全体における差分確率・線形確率はp^5となる。従って、データ変換装置全体では段数を３ｒとして確率$P \leqq p^{10r}$で近似表現でき、例えばｒ＝４（段数１２段）とすると$P \leqq p^{10}$である。これはＤＥＳ暗号に換算すると60段以上に相当し、差分解読法および線形解読法に対して十分安全なデータ変換装置となる。なお、第２の鍵依存線形変換部344 として図１７に示したもの以外の線形変換手段を用いることもできる。その例を図１８に示す。この場合は下記の演算がなされる。

$$mid_{10}=mid_{01}\oplus mid_{02}\oplus mid_{04}\oplus mid_{05}\oplus mid_{06}\oplus mid_{07}\oplus k_{i10} \quad (16\text{-}1)$$

$$mid_{11}=mid_{01}\oplus mid_{02}\oplus mid_{03}\oplus mid_{04}\oplus mid_{06}\oplus k_{i11} \quad (16\text{-}2)$$

$$mid_{12}=mid_{00}\oplus mid_{01}\oplus mid_{03}\oplus mid_{04}\oplus mid_{05}\oplus mid_{06}\oplus k_{i12} \quad (16\text{-}3)$$

$$mid_{13}=mid_{00}\oplus mid_{03}\oplus mid_{04}\oplus mid_{06}\oplus mid_{07}\oplus k_{i13} \quad (16\text{-}4)$$

$$mid_{14}=mid_{00}\oplus mid_{02}\oplus mid_{03}\oplus mid_{05}\oplus mid_{06}\oplus mid_{07}\oplus k_{i14} \quad (16\text{-}5)$$

$$mid_{15}=mid_{00}\oplus mid_{01}\oplus mid_{02}\oplus mid_{05}\oplus mid_{06}\oplus k_{i15} \quad (16\text{-}6)$$

$$mid_{16}=mid_{00}\oplus mid_{01}\oplus mid_{02}\oplus mid_{03}\oplus mid_{04}\oplus mid_{07}\oplus k_{i16} \quad (16\text{-}7)$$

$$mid_{17}=mid_{00}\oplus mid_{02}\oplus mid_{04}\oplus mid_{05}\oplus mid_{07}\oplus k_{i17} \quad (16\text{-}8)$$

あるいは、図１９に示す回路構成としてもよく、この場合は下記の演算がなされる。

$$mid_{10}=mid_{00}\oplus mid_{01}\oplus mid_{04}\oplus mid_{05}\oplus mid_{06}\oplus k_{i10} \quad (17\text{-}1)$$

$$mid_{11}=mid_{01}\oplus mid_{03}\oplus mid_{04}\oplus mid_{05}\oplus mid_{07}\oplus k_{i11} \quad (17\text{-}2)$$

$$mid_{12}=mid_{00}\oplus mid_{02}\oplus mid_{04}\oplus mid_{06}\oplus mid_{07}\oplus k_{i12} \quad (17\text{-}3)$$

$$mid_{13}=mid_{02}\oplus mid_{03}\oplus mid_{05}\oplus mid_{06}\oplus mid_{07}\oplus k_{i13} \quad (17\text{-}4)$$

$$mid_{14}=mid_{00}\oplus mid_{01}\oplus mid_{03}\oplus mid_{05}\oplus mid_{06}\oplus mid_{07}\oplus k_{i14} \quad (17\text{-}5)$$

$$mid_{15}=mid_{01}\oplus mid_{02}\oplus mid_{03}\oplus mid_{04}\oplus mid_{05}\oplus mid_{07}\oplus k_{i15} \quad (17\text{-}6)$$

$$mid_{16}=mid_{00}\oplus mid_{01}\oplus mid_{02}\oplus mid_{04}\oplus mid_{05}\oplus mid_{07}\oplus k_{i16} \quad (17\text{-}7)$$

$$mid_{17}=mid_{00}\oplus mid_{02}\oplus mid_{03}\oplus mid_{04}\oplus mid_{05}\oplus mid_{06}\oplus k_{i17} \quad (17\text{-}8)$$

第２鍵依存線形変換部344 は図１７～１９の演算から明らかなように、８つの入力データmid_{00}, mid_{01}, mid_{02}, …, mid_{07} のうち、それぞれ６つの入力データから生成した４つの出力データと、それぞれ５つの入力データから生成した４つの出力データの合計８つの出力データmid_{10}, mid_{11}, mid_{12}, …, mid_{17} を生成するような鍵依存線形変換であり、かつ８つの各入力データmid_{00}, mid_{01}, mid_{02}, …, mid_{07} が少なくとも４つ以上の他の系統の出力データに影響を与える（例えば図１７の例であれば、入力データmid_{00}は６つの出力データmid_{11}, mid_{12}, mid_{13}, mid_{14}, mid_{15}, mid_{17}に影響を与える）ような線形変換であれば、図１７の例について述べたと同様に、非線形関数部304 全体における差分確率・線形確率はp^5となる。

鍵データ{fk, k_{00}, k_{01}, k_{02}, k_{10}, k_{11}, k_{12}, … , $k_{(N-1)0}$, $k_{(N-1)1}$, $k_{(N-1)2}$, ek}は、主鍵情報ｋｅｙが入力部320 より鍵データ生成部321 に入力され、この鍵データ生成部321 によって変換され、鍵記憶部322 に蓄積されたデータである。

鍵データ生成部321 による鍵データの生成は図１に示したＤＥＳ暗号の拡大鍵生成ルーチン部２１と同様に構成してもよいし、あるいは米国特許No. 4, 850. 019に示されている拡大鍵生成部と同様に構成してもよい。

また、初期鍵依存変換部302 、最終鍵依存変換部308 、鍵依存線形変換部341, 344, 347は鍵に依存する線形変換手段であるため、差分解読法および線形解読法以外の解読法に対しても十分な安全性を兼ね備えたデータ変換装置である。

なお、第４実施例はこの例に特定されるだけでなく、例えば高速性を望むのであれば、これら初期鍵依存変換部302 、最終鍵依存変換部308 、鍵依存線形変換部341, 344, 347について、そのいずれかを削除する、または鍵に依存しない変換手段に変更することが可能である。この場合、差分解読法および線形解読法に対する安全性をそれほど低下させることなく、暗号化処理速度の向上が望める。

第５実施例

図１３に示した第４実施例と同様の構成を有するデータ変換装置において、非線形関数部304 の機能構成の変形例を用いた実施例を説明する。第５実施例の基

$$MID_{04}=[mid_{04}, 00\cdots0_{(2)}, mid_{04}, mid_{01}, mid_{04}, 00\cdots0_{(2)}, 00\cdots0_{(2)}, mid_{04}] \quad (18\text{-}5)$$

$$MID_{05}=[mid_{05}, mid_{05}, 00\cdots0_{(2)}, mid_{05}, mid_{05}, mid_{05}, 00\cdots0_{(2)}, 00\cdots0_{(2)}] \quad (18\text{-}6)$$

$$MID_{06}=[mid_{06}, mid_{06}, mid_{06}, 00\cdots0_{(2)}, 00\cdots0_{(2)}, mid_{06}, mid_{06}, 00\cdots0_{(2)}] \quad (18\text{-}7)$$

$$MID_{07}=[00\cdots0_{(2)}, mid_{07}, mid_{07}, mid_{07}, 00\cdots0_{(2)}, 00\cdots0_{(2)}, mid_{07}, mid_{07}] \quad (18\text{-}8)$$

これらのデータMID_{00}～MID_{07}は、第２実施例において式(8-1)～(8-4)について説明したと同様な手法により予め決めることができる。即ち、データMID_{00}は、図１７に示す線形変換部344Aにおいて、データmid_{00} 以外の全てのデータmid_{01}～mid_{07}をそれぞれ"$00\cdots0_{(2)}$"と設定した場合に、線形変換部344Aの８系統60_0～60_7の出力に得られるデータのセットである。データMID_{01}は、データmid_{01}以外の全てのデータmid_{00}, mid_{02}～mid_{07}をそれぞれ"$00\cdots0_{(2)}$"と設定した場合に、線形変換部344Aの８系統60_0～60_7の出力に得られるデータのセットである。以下、データMID_{02}～MID_{07}についても同様である。これら非線形変換部$343_0'$～$343_7'$はそれぞれデータin_0～in_7をアドレスとし、データMID_{00}～MID_{07}を直接読み出す変換テーブルとしてメモリにより構成してもよい。

次いで、図２１に示すように、データMID_{00}～MID_{07}は鍵データk_{i1}による第２鍵依存線形変換部344 に入力される。第２鍵依存線形変換部344 は、入力データを２つずつ互いに排他的論理和をとるXOR回路41_1～41_4と、それらの出力を２つずつ互いに排他的論理和をとるXOR回路42_1, 42_2と、それらの排他的論理和をとるXOR回路４３と、その出力と鍵データk_{i1}との排他的論理和をとるXOR回路４４とから構成されている。これにより、

$$MID_1=MID_{00}\oplus MID_{01}\oplus MID_{02}\oplus MID_{03}\oplus MID_{04}\oplus MID_{05}\oplus MID_{06}\oplus MID_{07}\oplus k_{i1} \quad (19)$$

が演算される。この出力MID_1は８個のブロックに分割され、データmid_{10}, mid_{11}, mid_{12}, …, mid_{17}として出力される。結局、第２鍵依存線形変換部344 による線形変換は、ｍビットのサブブロック単位として表すと次のようになる：

$$mid_{10}=mid_{01}\oplus mid_{02}\oplus mid_{03}\oplus mid_{04}\oplus mid_{05}\oplus mid_{06}\oplus k_{i10} \quad (20\text{-}1)$$

$$mid_{11}=mid_{00}\oplus mid_{02}\oplus mid_{03}\oplus mid_{05}\oplus mid_{06}\oplus mid_{07}\oplus k_{i11} \quad (20\text{-}2)$$

$$mid_{12}=mid_{00}\oplus mid_{01}\oplus mid_{03}\oplus mid_{04}\oplus mid_{06}\oplus mid_{07}\oplus k_{i12} \quad (20\text{-}3)$$

$$mid_{13}=mid_{00}\oplus mid_{01}\oplus mid_{02}\oplus mid_{04}\oplus mid_{05}\oplus mid_{07}\oplus k_{i13} \quad (20\text{-}4)$$

本構成は、図１３に示した第４実施例と同様である。異なる点は、その非線形関数部304 を示す図１４において、非線形変換部343_0～343_7の構成を、図８Ａ～８Ｄに示した第２実施例における非線形変換部$343_0'$, $343_1'$, $343_2'$, $343_3'$ 等と同様に変形し、拡大データを出力する構成とした点である。また、第２鍵依存線形変換部344 の構成を、図９に示したものと同様の構成としている。

図１３に示したように、右ブロックデータR_iは、鍵記憶部322 に蓄積されている鍵データk_{i0}, k_{i1}, k_{i2} と共に非線形関数部304 への入力データとなる。データR_iは、第１鍵依存線形変換部341 において、図１４で示したのと同様に鍵データk_{i0}と例えば排他的論理和がとられ、データ$R_i^* = R_i \oplus k_{i0}$ に線形変換される。次に、データR_i^*は分割部342 において８つのデータin_0, in_1, in_2, …, in_7 に分割される。８つのデータin_0, in_1, in_2, …, in_7 は、それぞれ非線形変換部$343_0'$, $343_1'$, $343_2'$, …, $343_7'$ において、データMID_{00}, MID_{01}, MID_{02}, …, MID_{07}に非線形変換される。非線形変換部$343_0'$は、ｍビットのデータin_0 を８×ｍビットのデータ

$$MID_{00} = [00\cdots0_{(2)}, mid_{00}, mid_{00}, mid_{00}, mid_{00}, mid_{00}, 00\cdots0_{(2)}, mid_{00}] \qquad (18\text{-}1)$$

に変換するよう構成されている。即ち、非線形変換部$343_0'$は、例えば図２０Ａに示すように、上位ｍビットに第４実施例の図１４における非線形変換部343_0と同じmid_{00}を出力し、下位ｍビットに"$00\cdots0_{(2)}$"を出力するS-box 343_0を有し、布線によりデータmid_{00}を６系統に分岐して出力すると共に、"$00\cdots0_{(2)}$"を２系統に分岐して出力する。

非線形変換部$343_1'$は図２０Ｂに示すように、上位ｍビットにデータmid_{01}を出力し、下位ｍビットに"$00\cdots0_{(2)}$"を出力するS-box を有し、布線によりデータmid_{01}を６系統に分岐して出力すると共に、ｍビットデータ"00…0"を２系統に分岐して出力する。以下、非線形変換部$343_2'$～$343_7'$も同様に構成され、図２０Ｃに非線形変換部$343_7'$の構成を示すが、その説明は省略する。これらの非線形変換部$343_1'$～$343_7'$は、それぞれデータin_1～in_7を以下のデータMID_{01}～MID_{07}に変換する。

$$MID_{01} = [mid_{01}, 00\cdots0_{(2)}, mid_{01}, mid_{01}, mid_{01}, mid_{01}, mid_{01}, 00\cdots0_{(2)}] \qquad (18\text{-}2)$$

$$MID_{02} = [mid_{02}, mid_{02}, 00\cdots0_{(2)}, mid_{02}, 00\cdots0_{(2)}, mid_{02}, mid_{02}, mid_{02}] \qquad (18\text{-}3)$$

$$MID_{03} = [mid_{03}, mid_{03}, mid_{03}, 00\cdots0_{(2)}, mid_{03}, 00\cdots0_{(2)}, mid_{03}, mid_{03}] \qquad (18\text{-}4)$$

$$mid_{11}=mid_{00}\oplus mid_{01}\oplus mid_{03}\oplus mid_{04}\oplus mid_{05}\oplus k_{i14} \quad (20\text{-}5)$$

$$mid_{15}=mid_{00}\oplus mid_{01}\oplus mid_{02}\oplus mid_{05}\oplus mid_{06}\oplus k_{i15} \quad (20\text{-}6)$$

$$mid_{16}=mid_{01}\oplus mid_{02}\oplus mid_{03}\oplus mid_{06}\oplus mid_{07}\oplus k_{i16} \quad (20\text{-}7)$$

$$mid_{17}=mid_{00}\oplus mid_{02}\oplus mid_{03}\oplus mid_{01}\oplus mid_{07}\oplus k_{i17} \quad (20\text{-}8)$$

これは図１７で説明した線形変換を表す式(15-1)～(15-8)と等価な線形変換である。これにより、第４実施例と同じデータmid_{10}, mid_{11}, mid_{12}, …, mid_{17}を生成する。なお、副鍵データk_{i1}は８つのデータk_{i10}, k_{i11}, k_{i12}, …, k_{i17}で構成されている。

次いで、データmid_{10}, mid_{11}, mid_{12}, …, mid_{17}は、それぞれ図１４中の非線形変換部345_0, 345_1, 345_2, …, 345_7において、データout_0, out_1, out_2, …, out_7に非線形変換された後、結合部346 において、８つのデータout_0, out_1, out_2, …, out_7が１つのデータY_i^*に結合される。最後に、データY_i^*は、鍵データk_{i2}による第３鍵依存線形変換部347 において、例えばk_{i2}ビット左ローテーションによりデータY_iに線形変換され、非線形関数部304 からの出力データY_iが生成される。

図２１に示すように、第２鍵依存線形変換部344 では、排他的論理和の個数が８つで図17と等価な線形変換（図１７では排他的論理和の個数は２４個）を実現しているため、第４実施例より高速な変換が可能になる。

また、第４実施例と同様に、非線形変換部$343_0'$～$343_7'$と345_0～345_7は８つずつ並列に配置されており、その非線形変換処理は相互に関連していないため、これらはすべて並列実行が可能である。さらに、非線形変換部$343_0'$～$343_7'$の差分確率・線形確率をｐとすると、非線形関数部304 全体における差分確率・線形確率はp^5となる。

なお上述において、第２（鍵依存）線形変換部344 は鍵k_{i1}ことなく、この入力サブデータ内の排他的論理和等によってもよい。つまり図１７のXOR63_0～63_7、これと対応する図１８，１９，２１路部分を省略してもよい。

また、上述において、第１鍵依存線形変換部341，第２鍵依存線形変換部344 及び第３鍵依存線形変換部347 は必ずしも鍵依存とすることはなく、つまり、鍵データを入力すくことなく、サブデータ内での線形変換を行ってもよい。

上述の第４及び第５実施例によるデータ拡散処理は、その処理手順を表すプログラムをコンピュータで実行することにより実現してもよい。その場合の処理手順は図１１及び１２に示すものと同様なので説明を省略する。

図２２は、第１～第５実施例で説明したデータ拡散処理手順を表すプログラムを予め記録媒体に記録しておき、そのプログラムを読み出してこの発明のデータ拡散を実施する構成例を示す。中央演算装置（ＣＰＵ）110 、リードオンリーメモリ（ＲＯＭ）120, ランダムアクセスメモリ（ＲＡＭ）130, 記憶装置（例えばハードディスクＨＤ）140, I/0インタフェース150, 及びこれらを互いに接続するバス160 は一般的なコンピュータ100 を構成している。この発明によるデータ拡散処理を実施するプログラムは、例えば記録媒体としてのハードディスク140 に格納されている。ＲＯＭ120 にはそれぞれのS-box が表として格納されている。データ拡散処理を実行する際に、ＲＡＭ130 にそのプログラムをハードディスク140 から読み込み、インタフェース150 を介して平文Ｍが入力されるとＣＰＵ110 の制御のもとにプログラムが実行され、生成された拡散データＣがインタフェース150 を介して出力される。

データ拡散処理を実行するプログラムは記録媒体としての任意の外部記憶装置180 に格納したものを使用してもよい。その場合、ドライバ170 を介して外部記憶装置180 からプログラムをハードディスク140 に一旦移して使用するか、ＲＡＭ130 に転送して使用することができる。

出力された拡散データＣは、図示してないが、例えば通信回線やインターネットを介して送出されると、共通鍵を有している者のみがこの拡散データＣを復号することができる。この発明により拡散されたデータＣは差分解読法及び線形解読法に対し強い耐性を有しているので、より安全な情報の送信が実現できる。

ところで、前述の各実施例において、鍵生成スケジュール部を図３と同様に構成すると、データ拡散部１０でk_iとk_{i+1} として使用する副鍵が、鍵スケジュール部２０中の鍵拡散部21_j の出力Q_{2j}とQ_{2j+1} （ただしi=2jとする）になっていた。一方、差分解読法や線形解読法で求まる可能性の高い副鍵はk_iとk_{i-1}であるから、これらの情報を用いてデータ拡散部を組み合わせることで、他の副鍵を求めやすくなっ

ていた。

そこで、前述の各実施例を代表する図４のデータ変換装置において、以下の実施例では副鍵を生成する鍵生成スケジュール部２０で実施される鍵生成アルゴリズムを、より複雑にすることでこの問題を解決する。以下の実施例では、少なくとも、副鍵k_iとk_{i-1}が求まっても、他のデータ拡散部の出力に関する多くの情報が洩れないようにするために、図３に示した鍵拡散部22（図３中の関数f_k）と同様の働きをするＧ関数部を用いることのほかに、鍵生成についての第１の観点によれば、Ｇ関数部の出力であるＬ成分が記憶部に一旦記憶され、必要な個数だけＬ成分を求めた後に、それぞれのＬ成分から出来るだけ均一に必要となる情報を抽出して副鍵を生成するデータ抽出機能を備えたＨ関数部が設けられる。また、第２の観点によれば、Ｇ関数部の出力であるＬ成分からそれぞれの副鍵として使用される部分情報がＨ関数部で抽出され、記憶部に記憶され、必要な個数のＬ成分から部分情報を抽出することで副鍵が生成される。

ＤＥＳの場合、主鍵のビット位置を入れ換えるだけで副鍵を生成していたので、鍵スケジュール処理は高速であった。しかし、副鍵の部分情報が知られると、直ちに主鍵の対応する情報が分かってしまう問題があった。

主鍵と副鍵の関係を複雑にするために、鍵スケジュールの処理量が大幅には増加しないように、かつ鍵スケジュール部のプログラム規模が増加しないようにするために、データ拡散部で使用するＦ関数あるいはＦ関数を構成するサブルーチン（以下ではこれらの関数をｆと書く）を利用することでデータ拡散関数Ｇ関数を構成して、Ｇ関数を繰り返し呼び出す、つまり繰り返し用いることで、複数個の中間値Ｌを生成する。

Ｇ関数は、２つの入力成分(Y, v)によって動作し、３つの出力成分(L, Y, v)を生成することとする。Ｙ成分のビット数は主鍵Ｋのビット数と一致するか、それよりも大きいものとする。

Ｇ関数は、データ拡散部に副鍵を供給するために、必要な回数（Ｍ回）繰り返し呼び出されて、Ｍ個のＬ成分を生成する（$0 \leqq j \leqq M-1$）。ｊ回目に呼び出されたＧ関数の出力を（L_j, Y_j, v_j）と表すと、この値の一部は(j+1)回目に呼び出さ

れるG関数の入力（$Y_{j+1}=Y_j$，$v_{j+1}=v_j$）として利用される。ここで、Y_0はその一部にKを含んだ値であり、v_0は予め定められた値（例えば0）とする。

与えられた主鍵Kに対し、副鍵k_i（$i=0,1,2,\cdots,N-1$）を以下のように定める。

$$(L_1,\ (Y_1,\ v_1))=G(Y_0,\ v_0) \qquad (21)$$

$$(L_{j+1},\ (Y_{j+1},\ v_{j+1}))=G(Y_j,\ v_j)$$
$$(j=1,\ 2,\ \cdots,\ M-1) \qquad (22)$$

$$k_i=H(i,\ L_1,\ L_2,\ \cdots,\ L_M)$$
$$(i=0,\ 1,\ 2,\ \cdots,\ N-1) \qquad (23)$$

ここで、H関数は、副鍵の添字iおよび関数Gの出力であるM個のL成分を入力して、それぞれのL_jから必要に応じてiによって決められたビット位置の情報を抽出するものである。

第６の実施例

図２３Ａに図４で示した鍵生成スケジュール部２０に適用する第６実施例の鍵生成スケジュール部の原理構成を示す。主鍵Kは中間鍵生成部220 に入力され、中間鍵生成部220 は縦続的に動作する複数（M段）のG関数部を有し、中間鍵L_1～L_Mを生成し、これら中間鍵は記憶部230 に記憶される。記憶部230 に記憶された中間鍵L_1～L_Mは副鍵生成部240 でH関数部にもとづき副鍵k_iが生成される。各部の構成作用を以下に具体的に説明する。

この例では、先に指摘した宮口らの米国特許で示されているデータランダム化部を利用することを想定して、先に示した図８の鍵スケジュール部の安全性を高める装置である。宮口らの米国特許で示されている鍵スケジュール部（図３）でN=16の場合にこの実施例を適用する場合について説明する。

図３では８(=N/2)段のデータ拡散部によって１６個のQ成分を入手する。ここでは、それぞれのQ成分をQ_jと表わす。各Q_jは１６ビットである。副鍵生成部240 ではそれぞれのQ_j成分の第１ビット目の値から副鍵k_0を構成し、それぞれのQ_j成分の第２ビット目の値から副鍵k_1を構成し、一般にそれぞれのQ_j成分の第ｉビット目の値から副鍵k_{i-1}を構成する。すなわち、Q_j成分の第ｉビット目を$Q_j[i]$と

表すと、副鍵k_iは次式で示される。

$$k_{i-1} = (Q_1[i],\ Q_2[i],\ \cdots,\ Q_j[i],\ \cdots,\ Q_{16}[i]) \qquad (24)$$

ここで、$1 \leqq i,\ j \leqq 16$に注意。

先に示したG関数、H関数の枠組で、ここに述べた処理方法を見直すと以下のとおりである。ここで、Y_jは64ビットの値であり、Y_j^LはY_jの上位32ビットの値、Y_j^RはY_jの下位32ビットの値を表している。

入力$(Y_j,\ v_j)$に対するG関数の出力を

$$(L_{j-1},\ (Y_{j-1},\ v_{j-1})) = G(Y_j,\ v_j) \qquad (0 \leqq j \leqq 7) \qquad (25)$$

と表すと、その出力結果$(L_{j-1}, (Y_{j-1},\ v_{j-1}))$は次式により得る。

$$Y_{j-1}^L = Y_j^R \qquad (26)$$

$$Y_{j-1}^R = L_{j-1} = f_k(Y_j^L, Y_j^R \oplus v_j) \qquad (27)$$

$$v_{j-1} = Y_j^L \qquad (28)$$

副鍵k_iはiと$L_1 \sim L_8$の関数として次式で示される。

$$k_i = H(i,\ L_1,\ L_2,\ \cdots,\ L_8) \qquad (29)$$

このH関数は、各L_jをビットごとに分割して$(t_j^{(1)},\ t_j^{(2)}, \cdots,\ t_j^{(32)})$と表したとき、副鍵$k_i$を

$$k_i = (t_1^{(i)},\ t_1^{(16+i)},\ t_2^{(i)},\ t_2^{(16+i)},\ \cdots,\ t_8^{(i)},\ t_8^{(16+i)}) \quad (1 \leqq i \leqq 16) \qquad (30)$$

と構成する。

なおこの方法では、最大16個の副鍵が得られるので、宮口らの米国特許で示されている暗号アルゴリズムでは、8段のF関数から構成した場合にまで対応できる。

図23Aの中間鍵生成部220の構成を図24を用いて説明する。G関数部22-1～22-8が縦続的に構成され、その初段のG関数部22-1に主鍵KがY_0として、また定数v_0が入力され、各j段目のG関数部22-jは、Y_{j-1}とv_{j-1}が入力され、Y_{i-1}が攪乱されて、出力L_j，Y_j，v_jを出力し、L_jが中間鍵として出力され、Y_j，v_jが次段のG関数部22-(j+1)へ供給される。つまり$Y_0 = K$、$v_0 = 0$と設定した後にG関数部22を8回呼び出す。G関数部の構成を図25に示す。この構成に対し、以

下の処理をｊ＝０からｊ＝７まで繰り返す。

ステップ１：Y_j とv_i をＧ関数部22-(j+1)に入力すると、図２５中のデータ分割装置221 を用いてY_j を２つのブロック(Y_j^L, Y_j^R)に分割する。

ステップ２：Y_j^L をv_{j+1} として出力する。また、Y_j^L をデータ拡散部(f_k)222 に入力する。

ステップ３：Y_j^R をデータ入換器224 に入力する。また、Y_j^R とv_j を排他的論理和回路223 に入力して$Y_j^R \oplus v_j$ を演算し、その結果をデータ拡散部(f_k)222 に入力する。

ステップ４：Y_j^L と$Y_j^R \oplus v_j$ を入力として受けとるとデータ拡散部f_k 222 は、その計算結果をL_{j+1} として出力すると同時に、データ入換器224 に入力する。

ステップ５：データ入換器224 は、Y_j^R とデータ拡散部(f_k) 222 の計算結果L_{j+1} を入力として受けとると、Y_j^R をY_{j+1}^L 、L_{j+1} をY_{j+1}^R とし、$Y_{j+1} = (Y_{j+1}^L, Y_{j+1}^R)$ と連結して出力する。

Ｇ関数部22-1～22-8が出力した８個のL_j を、一旦、記憶部230 （図２３Ａ）に記憶する。

次に、副鍵生成部240 としてのＨ関数部の構成を図２６を用いて説明する。Ｈ関数部240 は、記憶部230 から８個のＬ成分L_1 ～L_8 を読みだしたのちに以下を実行する。

ステップ１：記憶部230 から各L_j を読みだしてビット分割器241 に入力してそれぞれ、各１ビットずつ

$$(t_j^{(1)}, t_j^{(2)}, \cdots, t_j^{(32)}) = L_j \quad (j=1, 2, \cdots, 8) \qquad (31)$$

に分割する。

ステップ２：$(t_1^{(i)}, t_1^{(16+i)}, t_2^{(i)}, t_2^{(16+i)}, \cdots, t_8^{(i)}, t_8^{(16+i)})$ をビット結合器242 に入力して副鍵

$$k_i = (t_1^{(i)}, t_1^{(16+i)}, t_2^{(i)}, t_2^{(16+i)}, \cdots, t_8^{(i)}, t_8^{(16+i)})$$
$$(i=1, 2, \cdots, 16) \qquad (32)$$

を得る。

第７の実施例

上記第６実施例と同じ副鍵を出力する別の実施例を図２３Ｂ、図２４、図２５、図２７を参照して説明する。

図２３Ｂに示すように、中間鍵生成部220 で複数の中間鍵L_jが生成される。中間鍵生成部220 は図２３Ａのそれと同様であり、つまり図２４に示したように複数のＧ関数部22よりなる。このＧ関数部22で中間鍵L_jが生成されるごとに、その中間鍵L_jは、副鍵生成部250 で、副鍵k_iのｉとk_iのビット位置ｑとにより決まるビット位置の情報が、k_iのビット位置ｑの情報k_{iq}として選出され、記憶部260 に記憶される。

つまり中間鍵生成部220 と副鍵生成部250 とにより以下のステップ１からステップ７までを、j=0からj=7まで繰り返すことになる。

ステップ１：Y_jとv_jをＧ関数部22-(j+1)に入力すると、データ分割装置221 を用いてY_jを２つのブロック(Y_j^L, Y_j^R)に分割する。

ステップ２：Y_j^Lをv_{j+1}として出力する。また、Y_j^Lをデータ拡散部(fk)222 に入力する。

ステップ３：Y_j^Rをデータ入換器224 に入力する。また、Y_j^Rとv_jを排他的論理和回路223 に入力して$Y_j^R \oplus v_j$を入手後にデータ拡散部(f_k)222 に入力する。

ステップ４：Y_j^Lと$Y_j^R \oplus v_j$を入力として受けとるとデータ拡散部(f_k)222 は、その計算結果をL_{j+1}として副鍵生成部250 （図２３Ｂ）に入力すると同時に、データ入換器224 に入力する。

ステップ５：データ入換器224 は、Y_j^Rとデータ拡散部(f_k)222 の計算結果L_{j+1}を入力として受けとると、Y_j^RをY_{j+1}^L、L_{j+1}をY_{j+1}^Rとして$Y_{j+1} = (Y_{j+1}^L, Y_{j+1}^R)$と連結して出力する。

ステップ６：副鍵生成部250 は図２７に示すようにL_jをビット分割器251 に入力して下記のように１ビットごとに

$$(t_j^{(1)}, t_j^{(2)}, \cdots, t_j^{(32)}) = L_j \quad (j = 1, 2, \cdots, 8) \qquad (33)$$

に分割して、情報分配器252 に入力する。

ステップ７：情報分配器252 に入力されたビット列$(t_j^{(1)}, t_j^{(2)}, \cdots, t_j^{(32)})$はパラメータｉに対して$k_i$のビット位置ｑにより決まる$L_j$のビット位置の情報が、$k_i$

のビット位置qの情報とされ、

$$k_i = (t_1^{(i)},\ t_1^{(16+i)},\ t_2^{(i)},\ t_2^{(16+i)},\ \cdots,\ t_8^{(i)},\ t_8^{(16+i)}) \qquad (34)$$

となるように、副鍵k_i ごとに１６個に分割された記憶部260 にL_j ごとに記憶する。

ステップ８：それぞれのk_i に１６ビットの情報が設定されると、つまり副鍵k_iが生成されるとその値を出力する(i＝1, 2, …, 16)。

第８実施例

この実施例では鍵スケジュールを構成する際に、装置規模またはプログラムステップ数削減のため、暗号化で用いられている関数 f を用いる方法を説明する。

この例でも先に示したＧ関数、Ｈ関数の枠組で、説明する。

Ｇ関数に入力$(Y_j,\ v_j)$が与えられたときの出力を

$$(L_{j+1},\ (Y_{j+1},\ v_{j+1})) = G(Y_j,\ v_j) \qquad (0 \leqq j \leqq 7)$$

と表し、その出力を

$$((Y_j^{(1)},\ Y_j^{(2)},\ Y_j^{(3)},\ Y_j^{(4)}),\ v_j) \rightarrow$$

$$((L_{j+1}^{(1)}, L_{j+1}^{(2)}, L_{j+1}^{(3)}, L_{j+1}^{(4)}), [(Y_{j+1}^{(1)}, Y_{j+1}^{(2)}, Y_{j+1}^{(3)}, Y_{j+1}^{(4)}), v_{j+1}]) \qquad (35)$$

とする。ここで、

$$Y_{j+1}^{(i)} = f(Y_j^{(i)}) \quad (i=1,\ 2,\ 3,\ 4) \qquad (36)$$

$$L_{j+1}^{(0)} = v_j \qquad (37)$$

$$L_{j+1}^{(i)} = f(L_{j+1}^{(i-1)}) \oplus Y_{j+1}^{(i)} \quad (i=1,\ 2,\ 3,\ 4) \qquad (38)$$

$$v_{j+1} = L_{j+1}^{(4)} \qquad (39)$$

と定める。また、

$$k_i = H(i,\ L_1,\ L_2,\ \cdots,\ L_8) \qquad (40)$$

においては、

$$q_{i+4j} = L_{j+1}^{(i+1)} \quad (i=0,\ 1,\ 2,\ 3) \qquad (41)$$

$$(t_i^{(0)}, t_i^{(1)}, \ldots, t_i^{(7)}) = q_i \quad (i=0,\ 1,\ \cdots,\ 31) \qquad (42)$$

$$k_{(i+1)} = \left(t_{0+(i \bmod 2)}^{([i/2])}, t_{2+(i \bmod 2)}^{([i/2])}, \cdots, t_{30+(i \bmod 2)}^{([i/2])}\right) \quad (i = 0, 1, \cdots, 15) \quad (43)$$

と定める。ただし、式(43)における[i/2]は⌊i/2⌋を表すものとする。

この手順を図２８及び図２６を使って説明する。

準備

ステップ１：fのビット幅だけ、0123456789abcdef101112....(hex) の上位から同一ビット数取りだした値をv_0 として設定する。

ステップ２：主鍵KをY_0 に設定する。

中間鍵生成　以下の手順をj＝0, 1, 2, …, 7について繰り返す。

ステップ１：入力Y_j を４つに等分割し($Y_j^{(1)}$, $Y_j^{(2)}$, $Y_j^{(3)}$, $Y_j^{(4)}$)とする。

ステップ２：データ拡散器611～614を用いi＝1, 2, 3, 4 に対して、$Y_{j+1}^{(i)} = f(Y_j^{(i)})$をそれぞれ算出する。

ステップ３：v_j と$L_{j+1}^{(0)}$ を同一視する。

ステップ４：データ拡散器621～624を用いi＝1, 2, 3, 4, に対して、$f(L_{j+1}^{(i-1)})$を演算し、その演算結果を排他的論理和回路63i に入力して$Y_{j+1}^{(i)}$との排他的論理和演算を行って$L_{j+1}^{(i)} = f(L_{j+1}^{(i-1)}) \oplus Y_{j+1}^{(i)}$ を得る。

ステップ５：Y_{j+1} と($Y_{j+1}^{(1)}$, $Y_{j+1}^{(2)}$, $Y_{j+1}^{(3)}$, $Y_{j+1}^{(4)}$)を同一視する。

ステップ６：L_{j+1} と($L_{j+1}^{(1)}$, $L_{j+1}^{(2)}$, $L_{j+1}^{(3)}$, $L_{j+1}^{(4)}$)を同一視する。

ステップ７：$v_{j+1} = L_{j+1}^{(4)}$ とおく。

副鍵生成：　第６実施例と同様にして、式(43)を実現し、k_1, k_2, …, k_N ($N \leqq 16$)を得る。

なお、上記実施例は上記例に囚われることなく

(1) Y_0 のサイズが、Kより大きい場合は、Y_0 の一部をKとし、残りを定数で埋める。

(2) v_0 として、任意の定数とする。

(3) それぞれの文字のビット幅を整合性が取れる範囲で任意に設定する。

(4) fを暗号化に用いるのに利用した関数以外のものを使う。

(5) Hを計算する際にL_i の一部を使わない、つまり副鍵k_i の数が少なく、L_j の

ビット数が多い場合は、そのような状態になる。

(6) Hを計算する際に、第6実施例と同様のものを使う。

(7) Gを計算する際に第6実施例と同様のものを使う。

(8) 第7実施例と同様にして、中間鍵の全てを生成することなく1つの中間鍵が生成されるごとに記憶部260 のk_i の対応ビット位置に計算結果を格納する。

などとしても実行可能である。

中間鍵生成部220 、副鍵生成部240 、250 は図22に示したコンピュータによりプログラムを読出し実行するようにしてもよい。

発明の効果

以上、詳細に説明したように、この発明によれば、データを秘匿するための暗号化装置に用いるためのデータ変換装置について、安全性と高速性を同時に満たすような構造にすることによって、段数を大幅に増加させることなく安全性を確保し、かつ高速な暗号化処理が可能となるようなデータ変換装置を提供することができる。これにより、秘密鍵の制御のもとでデータをブロック単位で暗号化または復号を行う共通鍵暗号方式による暗号化装置に有用である。

また、この発明による鍵生成スケジュールによれば、第6および第7の実施例では、k_6, k_7, k_8, k_9, k_{10}, k_{11}が判明した場合でも、それぞれのL_j 成分の12ビット分（例えば、第6, 7, 8, 9, 10, 11, 22, 23, 24, 25, 26, 27ビット目）が求まっただけであるから、DESおよび宮口らの米国特許で示されている鍵スケジュール部の安全性に関する問題は解決されている。

付録

表記

#{a|cond} ：条件condを満たす集合 a の個数。

Δx, Γx ： x の差分値、 x のマスク値。

a•b ： a と b のビット毎の論理積に対する偶数パリティ値。

線形変換部 ： $P=[t_{ij}]$, $t_{ij}\in\{0, 1\}$, $0 \leq i, j \leq n-1$

線形変換部の入力：$z={}^T[z_0, \cdots, z_{n-1}]$, $z_i \in GF(2^a)$, $0 \le i \le n-1$

線形変換部の出力：$z'={}^T[z'_0, \cdots, z'_{n-1}]=P_z$, $z'_i \in GF(2^a)$, $0 \le i \le n-1$

定義

[定義１] Δx, Δy, Γx, Γy が与えられたとき、s-box の差分確率$DP^s(\Delta x \rightarrow \Delta y)$と線形確率$LP^s(\Gamma y \rightarrow \Gamma x)$を以下のように定める：

$$DP^s(\Delta x \rightarrow \Delta y) = \#\{x \in GF(2^a) \mid s(x)+s(x+\Delta x) = \Delta y\}/2^a$$

$$LP^s(\Gamma y \rightarrow \Gamma x) = (2 \times \#\{x \in GF(2^a) \mid x \bullet \Gamma x = s(x) \bullet \Gamma y\}/2^a - 1)^2$$

[定義２] s-box の最大差分確率p_sと最大線形確率q_sを以下のように定める。

$$p_s = \max_{\Delta x \neq 0, \Delta y} DP^s(\Delta x \rightarrow \Delta y)$$

$$q_s = \max_{\Gamma x, \Gamma y \neq 0} LP^s(\Gamma y \rightarrow \Gamma x)$$

[定義３] 差分解読法では入力差分値Δx が非零であるs-box 、線形解読法では出力マスク値Γy が非零であるs-box のことをactive s-box と呼ぶ。

[定義４] 2-round SPN構造で構成されるラウンド関数において、Δx, Δy, Γx, Γy が与えられたとき、差分確率$p(\Delta x \rightarrow \Delta y)$と線形確率$q(\Gamma y \rightarrow \Gamma x)$を以下のように定める。

$$p(\Delta x \rightarrow \Delta y) = \max_{\Delta z} \prod_{i=0}^{n-1} DP^s(\Delta x_i \rightarrow \Delta z_i) p(\Delta z \rightarrow \Delta z'_i) DP^s(\Delta z'_i \rightarrow \Delta y_i)$$

$$q(\Gamma y \rightarrow \Gamma x) = \max_{\Gamma z'} \prod_{i=0}^{n-1} LP^s(\Gamma y_i \rightarrow \Gamma z'_i) p(\Gamma z' \rightarrow \Gamma z_i) LP^s(\Gamma z_i \rightarrow \Gamma x_i)$$

ここで、$\Delta x=(\Delta x_0, \cdots, \Delta x_{n-1})$であり、$\Delta y$, Γx, Γy も同様である。更に、$p(\Delta z \rightarrow \Delta z'_i)$は、各 i に対して、$\Delta z$ が線形変換部によって$\Delta z'_i$に変換されることを表しているものとする。即ち、全ての i に対して、$\exists \Delta z'_i$ s.t. $p(\Delta z \rightarrow \Delta z'_i) = 1$である。$p(\Gamma z' \rightarrow \Gamma z_i)$も同様である。

[定義５] 2-round SPN構造で構成されるラウンド関数の最大差分確率 p と最大線形確率 q を以下のように定める。

$$p = \max_{\Delta x \neq 0, \Delta y} p(\Delta x \rightarrow \Delta y)$$

$$q = \max_{\Gamma x, \Gamma y \neq 0} q(\Gamma y \rightarrow \Gamma x)$$

［定義６］ 2-round SPN構造に対して、全てのs-box が全単射である時、差分解読法と線形解読法におけるactive s-box の最少個数n_d, n_lは以下のように与えられる。

$$n_d = \min_{\Delta z \neq 0} [H_w(\Delta z) + H_w(\Delta z')]$$

$$n_l = \min_{\Gamma z' \neq 0} [H_w(\Gamma z) + H_w(\Gamma z')]$$

ここで、

$$H_w(z) = \#\{0 \leqq i < n \mid z_i \neq 0\}$$

とする。また、Δz, Γz は線形変換部の入力差分値と入力マスク値を表し、$\Delta z'$, $\Gamma z'$ は線形変換部の出力差分値と出力マスク値を表す。

［定理１］ 2-round SPN構造のラウンド関数における最大差分確率ｐと最大線形確率ｑは以下の関係式を満たす。

$$p \leqq p_s^{n_d} \quad \text{及び} \quad q \leqq q_s^{n_l}$$

[Proof sketch]

差分解読法の場合について考える。定義２と定義３より、active s-box における差分確率はp_s以下である。従って、あるΔx, Δy におけるactive s-box の個数をn_{dz} とすると、そのときのラウンド関数の差分確率は、定義４より、$p_s^{n_{dz}}$以下となる。一方、n_dをactive s-box の最少個数とすると、全てのΔx, Δy に対して$n_d \leqq n_{dz}$, $p_s \leqq 1$であるから、

$$\forall n_{dz}\ p_s^{n_{dz}} \leqq p_s^{n_d}$$

となる。従って、このラウンド関数における最大差分確率ｐは、$p \leqq p_s^{n_d}$である。同様のことが、線形解読法の場合についてもいえ、$q \leqq q_s^{n_l}$が導かれる。

［定理２］ Concatenation Rules

X⊣ (Y, Z)を、データＸがデータＹとデータＺの２つのデータに同じく複製されることを表すとする。即ち、X=Y=Z が成立する。このとき、以下の関係が成立

する：

$X=Y\oplus Z \Rightarrow \Delta X=\Delta Y\oplus\Delta Z,\ \Gamma X=\Gamma Y=\Gamma Z$ (XOR operation)

$X\dashv (Y,\ Z) \Rightarrow \Delta X=\Delta Y=\Delta Z,\ \Gamma X=\Gamma Y\oplus\Gamma Z$ (Branch operation)

[差分解読法及び線形解読法に対する耐性]

定理１の式で示したように、差分解読法及び線形解読法に対するラウンド関数の耐性は、active s-box の最少個数n_d, n_lから決めることができる。例えば、行列P_Eが表す線形変換は全単射であるので、$H_w(\Delta z)\geqq 2$ならば$H_w(\Delta z')\geqq 1$であり、また$H_w(\Delta z)=1$ならば$H_w(\Delta z')\geqq 3$であることから、active s-box の個数が２になることはあり得ないので、$n_d\geqq 3$となる。ところで、例えば、$\Delta z_0=\Delta z_2\neq 0$, $\Delta z_1=\Delta z_3=0$の時、$\Delta z'_0=\Delta z_2\neq 0$, $\Delta z'_1=\Delta z'_2=\Delta z'_3=0$ ($H_w(\Delta z')=1$)となり、active s-box の個数は、$H_w(\Delta z)+H_w(\Delta z')=3$である。従って、$n_d=3$となるので、定理１よりこのラウンド関数の耐性は$p\leqq p_s^3$である。また$n_l$についても同様の議論ができ、$n_l=3$, $q\leqq q_s^3$であることが示せる。

請求の範囲

1． 複数の鍵データを保持する鍵記憶手段と、上記複数の鍵データが与えられ、それぞれ鍵依存の非線形変換を行う非線形関数部をそれぞれ含む縦続接続された複数のラウンド処理部とを含み、入力データを鍵データに依存して別のデータに変換するデータ変換装置であり、各上記ラウンド処理部の上記非線形関数部は：

上記ラウンド処理部への入力データを、上記鍵記憶手段に蓄積された第１鍵データに基づいて線形変換を行う第１鍵依存線形変換手段と、

上記第１鍵依存線形変換手段の出力データをｎ個のサブデータに分割する分割手段と、ｎは４以上の正数であり、

上記ｎ個のサブデータのそれぞれに非線形変換を行う第１非線形変換手段と、

上記鍵記憶手段に蓄積された第２鍵データと、上記第１非線形変換手段の各々の出力サブデータとで線形変換を行う第２鍵依存線形変換手段と、

上記第２鍵依存線形変換手段のｎ個の出力サブデータのそれぞれに非線形変換を行う第２非線形変換手段と、及び

上記第２非線形変換手段のｎ個の出力サブデータを結合して上記非線形関数手段の出力とする結合手段、

とを含み、上記第２鍵依存線形変換手段は、その入力に対しｎ×ｎ行列で規定される排他的論理和を行う線形変換層を含んでいる。

2． 請求項１記載のデータ変換装置において、

上記入力データを二つの部分データに分割する初期分割手段と、

上記部分データの一つを入力とする上記非線形関数手段と、

上記非線形関数手段の出力データを残りの部分データの一つに作用させる線形演算手段と、

二つの部分データを一つの出力データに結合する最終結合手段、

とを備える。

3． 請求項２に記載のデータ変換装置において、

上記入力データに変換を行って上記初期分割手段へ供給する初期変換手段を備えることを特徴とするデータ変換装置。

4． 請求項２又は３に記載のデータ変換装置において、 上記最終結合手段の出力データに変換を行って上記データ変換装置の出力データとする最終変換手段を備えることを特徴とするデータ変換装置。

5． 請求項３又は４に記載のデータ変換装置において、 上記初期変換手段及び上記最終変換手段の少くとも一方は、上記鍵記憶手段に蓄積された鍵データに基づいて変換を行う鍵依存変換手段であることを特徴とするデータ変換装置。

6． 請求項１乃至５のいずれか１つに記載のデータ変換装置において、

上記非線形関数手段は、上記鍵記憶手段に蓄積された第３鍵データにより上記結合手段の出力データを線形変換して上記非線形関数手段の出力とする第３鍵依存線形変換手段を備えることを特徴とするデータ変換装置。

7． 請求項１乃至６のいずれか１つに記載のデータ変換装置において、

上記第１鍵依存線形変換手段、上記第２鍵依存線形変換手段、上記第３鍵依存線形手段のいずれかは、固定された線形変換を行う線形変換手段であることを特徴とするデータ変換装置。

8． 請求項１乃至７のいずれか１つに記載のデータ変換装置において、

上記第１非線形変換手段及び上記第２非線形変換手段は、その各入力サブデータを、二つのサブブロックに分割する手段と、その分割された二つのサブブロックに対し、それぞれ線形変換を行う手段と、非線形変換を行う手段とを縦続的に行う手段と、その縦続的に行う手段の各変換出力サブブロックを統合して対応入力サブデータに対する非線形変換出力サブデータとする手段を備えていることを特徴とするデータ変換装置。

9． 請求項１乃至８のいずれか１つに記載のデータ変換装置において、上記ｎ×ｎ行列は、予め決定した安全性閾値Ｔに対し、それぞれがハミング重みT-1 以上となるｎ個の列ベクトルから構成されている。

１０． 請求項９に記載のデータ変換装置において、上記行列は、上記ハミング重みがT-1 以上となる列ベクトルから構成された複数の行列の候補から、最大のn_dを与えるものとして選択されたものであり、上記n_dはactive s-boxの最少個数である。

１１．　請求項１乃至１０のいずれか１つに記載のデータ変換装置において、上記ｎ×ｎ行列は４×４行列である。

１２．　請求項１１に記載のデータ変換装置において、

上記第２線形変換手段は、上記第１非線形変換手段よりの４つの出力A1，A2，A3，A4を入力して、

$B1=A1\oplus A3\oplus A4$

$B2=A2\oplus A3\oplus A4$

$B3=A1\oplus A2\oplus A3$

$B4=A1\oplus A2\oplus A4$

をそれぞれ演算して、データB1，B2，B3，B4を出力する手段であることを特徴とするデータ変換装置。

１３．　請求項１２に記載のデータ変換装置において、

上記第２線形変換手段は鍵依存線形変換手段であって、上記鍵記憶手段よりの鍵データk2＝[k21，k22，k23，k24] も入力され、上記B1，B2，B3，B4を得る演算に、それぞれk21，k22，k23，k24も排他的論理和算されることを特徴とするデータ変換装置。

１４．　請求項１１に記載のデータ変換装置において、

上記第１非線形変換手段は上記分割手段よりの４個の各ｍビットのサブデータin1，in2，in3，in4のin1を4mビットのMI1＝[A1，$00\cdots0_{(2)}$，A1，A1]に変換する手段と、in2を4mビットのMI2＝[$00\cdots0_{(2)}$，A2，A2，A2]に変換する手段と、in3を4mビットのMI3＝[A3，A3，A3，$00\cdots0_{(2)}$]に変換する手段と、in4を4mビットのMI4＝[A4，A4，$00\cdots0_{(2)}$，A4]に変換する手段とよりなり、

上記第２線形変換手段は、上記第１非線形変換手段よりのデータMI1，MI2，MI3，MI4を入力して、$B=MI1\oplus MI2\oplus MI3\oplus MI4$を演算し、B＝[B1，B2，B3，B4]を出力する手段であることを特徴とするデータ変換装置。

１５．　請求項１４に記載のデータ変換装置において、

上記第２線形変換手段は鍵依存線形変換手段であって、上記鍵記憶手段よりの４ｍビットの鍵データk2も入力され、上記Ｂの演算にk2も排他的論理和算される

ことを特徴とするデータ変換装置。

１６．　請求項１乃至１０のいずれか１つに記載のデータ変換装置において、上記ｎ×ｎ行列は８×８行列である。

１７．　請求項１６に記載のデータ変換装置において、

上記第２線形変換手段は、その８つの出力サブデータB1～B8を、上記第１非線形変換手段の８つの出力サブデータA1，A2，…，A8のうち、６つのサブデータの排他的論理和算により４つの出力サブデータB1，B2，B3，B4を得、上記８つの出力サブデータ中の５つのサブデータの排他的論理和算により４つの出力サブデータB5，B6，B7，B8を得る手段であることを特徴とするデータ変換装置。

１８．　請求項１７に記載のデータ変換装置において、上記第２線形変換手段は鍵依存線形変換手段であって、上記鍵記憶手段に蓄積された鍵データk2＝[k21，k22，k23，k24，K25，k26，k27，k28]も入力され、上記出力サブデータ[B1，B2，B3，B4，B5，B6，B7，B8]を得る排他的論理和算にそれぞれ、鍵データk21，k22，k23，k24，k25，k26，k27，k28が排他的論理和されることを特徴とするデータ変換装置。

１９．　請求項１６に記載のデータ変換装置において、

上記第１非線形変換手段は、上記分割手段よりの８個の各ｍビットのサブデータin1～in8をそれぞれ8mビットのデータ

$MI1=[00\cdots0_{(2)}, A1, A1, A1, A1, A1, 00\cdots0_{(2)}, A1]$、

$MI2=[A2, 00\cdots0_{(2)}, A2, A2, A2, A2, A2, 00\cdots0_{(2)}]$、

$MI3=[A3, A3, 00\cdots0_{(2)}, A3, 00\cdots0_{(2)}, A3, A3, A3]$、

$MI4=[A4, A4, A4, 00\cdots0_{(2)}, A4, 00\cdots0_{(2)}, A4, A4]$、

$MI5=[A5, 00\cdots0_{(2)}, A5, A5, A5, 00\cdots0_{(2)}, 00\cdots0_{(2)}, A5]$、

$MI6=[A6, A6, 00\cdots0_{(2)}, A6, A6, A6, 00\cdots0_{(2)}, 00\cdots0_{(2)}]$、

$MI7=[A7, A7, A7, 00\cdots0_{(2)}, 00\cdots0_{(2)}, A7, A7, 00\cdots0_{(2)}]$、及び

$MI8=[00\cdots0_{(2)}, A8, A8, A8, 00\cdots0_{(2)}, 00\cdots0_{(2)}, A8, A8]$

に変換する手段であり、

上記第２線形変換手段は、上記第１非線形変換手段よりのデータMI1～MI8を入

力して、B＝MI1⊕MI2⊕MI3⊕MI4⊕MI5⊕MI6⊕MI7⊕MI8を演算し、B＝[B1, B2, B3, B4, B5, B6, B7, B8]を出力する手段であることを特徴とするデータ変換装置。

２０．　請求項１９に記載のデータ変換装置において、

上記第２線形変換手段は鍵依存線形変換手段であって、上記鍵記憶手段の8mビットの鍵データk2も入力され、上記Ｂの演算にk2も排他的論理和算されることを特徴とするデータ変換装置。

２１．　鍵記憶手段に保持された複数の鍵データが与えられ、それぞれ鍵依存の非線形変換を行う非線形関数処理を含むラウンド処理を複数回縦続して実行することにより、入力データを鍵データに依存して別のデータに変換するデータ変換プログラムが記録された記録媒体であり、各上記ラウンド処理の上記非線形関数処理は：

上記ラウンド処理部への入力データを、上記鍵記憶手段に保持された第１鍵データに基づいて線形変換を行う第１鍵依存線形変換ステップと、

上記第１鍵依存線形変換ステップによる出力データをｎ個のサブデータに分割する分割ステップと、ｎは４以上の整数であり、

上記ｎ個のサブデータのそれぞれに非線形変換を行う第１非線形変換ステップと、

第２鍵データと、上記第１非線形変換ステップの出力サブデータとで線形変換を行う第２鍵依存線形変換ステップと、

上記第２鍵依存線形変換ステップによるｎ個の出力サブデータのそれぞれに第２の非線形変換を行う第２非線形変換ステップと、及び

上記第２非線形変換ステップによるｎ個の出力サブデータを結合して上記非線形関数処理結果として出力する結合ステップ、

とを含み、

上記第２鍵依存線形変換ステップは、その入力に対しｎ×ｎ行列で規定される排他的論理和を行う排他的論理和線形変換ステップを含んでいる。

２２．　請求項２１に記載の記録媒体において、上記データ変換プログラムは、

上記入力データを二つの部分データに分割する初期分割ステップと、

上記部分データの一つを入力とする上記非線形関数処理を行うステップと、

上記非線形関数処理ステップの出力データを残りの部分データの一つに作用させる線形演算ステップと、

二つの部分データを一つの出力データに結合する最終結合ステップ、

とを含む。

２３．　請求項２２に記載の記録媒体において、上記データ変換プログラムは、上記入力データに変換を行って上記初期分割ステップへ供給する初期変換ステップを含む。

２４．　請求項２２又は２３に記載の記録媒体において、上記データ変換プログラムは　上記最終結合ステップの出力データに変換を行って上記データ変換出力データとする最終変換ステップを含む。

２５．　請求項２３又は２４に記載の記録媒体において、上記データ変換プログラムの、上記初期変換ステップ及び上記最終変換ステップの少くとも一方は、鍵データに基づいて変換を行う鍵依存変換ステップである。

２６．　請求項２１乃至２５のいずれか１つに記載の記録媒体において、

上記非線形関数処理ステップは、上記鍵保持手段に保持された第３鍵データにより上記結合ステップの出力データを線形変換して上記非線形関数処理ステップの出力とする第３鍵依存線形変換ステップを含む。

２７．　請求項２１乃至２８のいずれか１つに記載の記録媒体において、

上記第１鍵依存線形変換ステップ、上記第２鍵依存線形変換ステップ、上記第３鍵依存線形変換ステップのいずれかは、固定された線形変換を行う線形変換ステップである。

２８．　請求項２１乃至２７のいずれか１つに記載の記録媒体において、

上記第１非線形変換ステップ及び上記第２非線形変換ステップは、その各入力サブデータを、二つのサブブロックに分割するステップと、その分割された二つのサブブロックに対し、それぞれ線形変換を行う処理と、非線形変換を行う処理とを縦続的に行うステップと、その縦続的に行うステップの各変換出力サブブロックを統合して対応入力サブデータに対する非線形変換出力サブデータとするス

テップを含む。

２９．　請求項２１乃至２８のいずれか１つに記載の記録媒体において、上記ｎ×ｎ行列は、予め決定した安全性閾値Ｔに対し、それぞれがハミング重みT-1 以上となるｎ個の列ベクトルから構成されている。

３０．　請求項２９に記載の記録媒体において、上記行列は、上記ハミング重みがT-1 以上となる列ベクトルから構成された複数の行列の候補から、最大のn_dを与えるものとして選択されたものであり、上記n_dはactive s-boxの最少個数である。

３１．　請求項２１から３０のいずれかに記載の記録媒体において、上記ｎ×ｎ行列は４×４行列である。

３２．　請求項３１に記載の記録媒体において、

上記第２線形変換ステップは、上記第１非線形変換ステップよりの４つの出力A1，A2，A3，A4を入力して、

B1＝A1⊕A3⊕A4

B2＝A2⊕A3⊕A4

B3＝A1⊕A2⊕A3

B4＝A1⊕A2⊕A4

をそれぞれ演算して、データB1，B2，B3，B4を出力するステップである。

３３．　請求項３２に記載の記録媒体において、

上記第２線形変換ステップは鍵依存線形変換ステップであって、上記鍵記憶手段よりの鍵データk2＝[k21，k22，k23，k24] も入力され、上記B1，B2，B3，B4を得る演算に、それぞれk21，k22，k23，k24も排他的論理和算される。

３４．　請求項３２又は３３に記載の記録媒体において、

上記第１非線形変換ステップは上記分割ステップよりの４個の各ｍビットのサブデータin1，in2，in3，in4のin1を4mビットのMI1＝[A1，$00\cdots0_{(2)}$，A1，A1]に変換するステップと、in2を4mビットのMI2＝[$00\cdots0_{(2)}$，A2，A2，A2]に変換するステップと、in3を4mビットのMI3＝[A3，A3，A3，$00\cdots0_{(2)}$]に変換するステップと、in4を4mビットのMI4＝[A4，A4，$00\cdots0_{(2)}$，A4]に変換するステップとよりなり、

上記第２線形変換ステップは、上記第１非線形変換ステップよりのデータMI1, MI2, MI3, MI4を入力して、B＝MI1⊕MI2⊕MI3⊕MI4を演算し、B＝[B1, B2, B3, B4]を出力するステップである。

３５．　請求項３４に記載の記録媒体において、

上記第２線形変換ステップは鍵依存線形変換ステップであって、上記鍵記憶手段よりの4mビットの鍵データk2も入力され、上記Ｂの演算にk2も排他的論理和算される。

３６．　請求項２１乃至３０のいずれか１つに記載の記録媒体において、上記ｎ×ｎ行列は８×８行列である。

３７．　請求項３６に記載の記録媒体において、

上記第２線形変換ステップは、その８つの出力サブデータB1～B8を、上記第１非線形変換ステップの８つの出力サブデータA1, A2, …, A8のうち、６つのサブデータの排他的論理和算により４つの出力サブデータB1, B2, B3, B4を得、上記８つの出力サブデータ中の５つのサブデータの排他的論理和算により４つの出力サブデータB5, B6, B7, B8を得るステップである。

３８．　請求項３７に記載の記録媒体において、上記第２線形変換ステップは鍵依存線形変換ステップであって、上記鍵記憶手段に蓄積された鍵データk2＝[k21, k22, k23, k24, K25, k26, k27, k28]も入力され、上記出力サブデータ[B1, B2, B3, B4, B5, B6, B7, B8]を得る排他的論理和算にそれぞれ、鍵データk21, k22, k23, k24, k25, k26, k27, k28が排他的論理和される。

３９．　請求項３７又は３８に記載の記録媒体において、

上記第１非線形変換ステップは、上記分割ステップよりの８個の各ｍビットのサブデータin1～in8をそれぞれ8mビットのデータ

MI1＝[00…$0_{(2)}$, A1, A1, A1, A1, A1, 00…$0_{(2)}$, A1]、

MI2＝[A2, 00…$0_{(2)}$, A2, A2, A2, A2, A2, 00…$0_{(2)}$]、

MI3＝[A3, A3, 00…$0_{(2)}$, A3, 00…$0_{(2)}$, A3, A3, A3]、

MI4＝[A4, A4, A4, 00…$0_{(2)}$, A4, 00…$0_{(2)}$, A4, A4]、

MI5＝[A5, 00…$0_{(2)}$, A5, A5, A5, 00…$0_{(2)}$, 00…$0_{(2)}$, A5]、

MI6＝[A6, A6, 00…$0_{(2)}$, A6, A6, A6, 00…$0_{(2)}$, 00…$0_{(2)}$]、

MI7＝[A7, A7, A7, 00…$0_{(2)}$, 00…$0_{(2)}$, A7, A7, 00…$0_{(2)}$]、及び

MI8＝[00…$0_{(2)}$, A8, A8, A8, 00…$0_{(2)}$, 00…$0_{(2)}$, A8, A8]

に変換するステップであり、

上記第２線形変換ステップは、上記第１非線形変換ステップよりのデータMI1～MI8を入力して、

B＝MI1⊕MI2⊕MI3⊕MI4⊕MI5⊕MI6⊕MI7⊕MI8

を演算し、B＝[B1, B2, B3, B4, B5, B6, B7, B8]を出力するステップである。

４０．　請求項３９に記載の記録媒体において、

上記第２線形変換ステップは鍵依存線形変換ステップであって、上記鍵記憶手段の8mビットの鍵データk2も入力され、上記Ｂの演算にk2も排他的論理和算される。

４１．　請求項１乃至２０のいずれか１つのデータ変換装置において、更に、

主鍵Ｋが入力され、縦続的に動作するＭ段からなり、各段より中間値L_{j+1}（j＝0, 1, …, M-1）成分を生成するＧ関数手段と、

そのＧ関数手段の出力である各中間値L_j成分を一旦記憶しておくための中間値記憶手段と、

複数個のL_j成分からＮ個の副鍵を生成し、上記鍵記憶手段に上記複数の鍵データとして記憶する部分情報抽出機能を備えたＨ関数手段、

とを含み、

上記Ｇ関数手段は、上記主鍵Ｋを、Y_0の少なくとも一部として取り込み、j+1段目が（j＝0, 1, …, M-1）ｊ段目の出力（L_j, Y_j, v_j）中のY_jとv_jを入力し、その入力を拡散して、L_{j+1}, Y_{j+1}, v_{j+1}を出力し、をＫとする手段であり、

上記Ｈ関数手段は、ｉ（i＝1, 2, …, N）と上記中間値記憶手段のL_1, L_2, …, L_Mとを入力とし、ｉによって決められたビット位置の情報をL_1, …, L_Mより抽出する副鍵k_iを出力し、上記副鍵は上記複数の鍵データとして上記鍵記憶手段に蓄積される。

４２．　請求項１乃至２０のいずれか１つに記載のデータ変換装置において、更

に、

主鍵Kが入力され、縦続的に動作するM段からなり、各段より中間値L_j（j＝0, 1, …, M-1）成分を生成するG関数手段と、

そのG関数手段で生成された複数個のL_j成分から副鍵を生成する部分情報抽出機能を備えたH関数手段と、

そのH関数手段の出力を副鍵k_iに対応する値として記憶しておくための中間値記憶手段、

とを含み、

上記G関数手段は、上記主鍵Kを、Y_0の少なくとも一部として取り込み、j+1段目がj段目の出力（L_j, Y_j, v_j）中のY_j, v_jを入力し、その入力を拡散してL_{j+1}, Y_{j+1}, v_{j+1}を出力する手段であり、

上記H関数手段は、i, q, L_j（$1 \leqq i \leqq N$, $1 \leqq j \leqq M$, $1 \leqq q \leqq k_i$のビット数）を入力として、L_jから、iとqによって決められたビット位置情報を抽出して、副鍵k_iのビット位置qの情報とする手段であり、上記副鍵は上記複数の鍵データとして上記鍵記憶手段に記憶される。

４３．請求項４１又は４２に記載のデータ変換装置において、上記G関数手段は、入力されたY_jを２つのブロック(Y_j^L, Y_j^R)に分割し、Y_j^Lをv_{j+1}として出力するデータ分割手段と、

上記Y_j^Rと上記v_jとから$Y_j^R \oplus v_j$を演算する排他的論理和手段と、

上記Y_j^Lと上記排他的論理和手段の出力とがあたえられ、それらを互いに拡散した結果をL_{j+1}しとして出力するデータ拡散手段と、

上記Y_j^RをY_{j+1}^Lとし、上記L_{j+1}をY_{j+1}^Rとし、これらY_{j+1}^LとY_{j+1}^Rを互いに連結して$Y_{j-1}=(Y_{j+1}^L, Y_{j+1}^R)$として出力するデータ入れ換え手段、

とを含む。

４４．請求項４１に記載のデータ変換装置において、上記H関数手段は、

上記中間値記憶手段から読み出した各L_jを各１ビットずつ

$$(t_j^{(1)}, t_j^{(2)}, \cdots, t_j^{(2N)}) = L_j \quad (j=1, 2, \cdots, M)$$

に分割するビット分割手段と、

得られた$(t_1^{(i)}, t_1^{(N+i)}, t_2^{(i)}, t_2^{(N+i)}, \cdots, t_M^{(i)}, t_M^{(N+i)})$を結合して副鍵

$$k_i=(t_1^{(i)}, t_1^{(N+i)}, t_2^{(i)}, t_2^{(N+i)}, \cdots, t_M^{(i)}, t_M^{(N+i)}) \quad (i=1, 2, \cdots, N)$$

を出力するビット結合手段、

とを含む。

４５．請求項４２に記載のデータ変換装置において、上記H関数手段は、

上記各L_jを各１ビットずつ

$$(t_j^{(1)}, t_j^{(2)}, \cdots, t_j^{(2N)})= L_j \quad (j=1, 2, \cdots, M)$$

に分割するビット分割手段と、

上記ビット$(t_j^{(1)}, t_j^{(2)}, \cdots, t_j^{(2N)})$から i に対して$k_i$のビット位置 q により決まる$L_j$のビット位置の情報が、$k_i$のビット位置となるよう結合して副鍵

$$k_i=(t_1^{(i)}, t_1^{(N+i)}, t_2^{(i)}, t_2^{(N+i)}, \cdots, t_M^{(i)}, t_M^{(N+i)}) \quad (i=1, 2, \cdots, N)$$

を出力するビット結合手段、

とを含む。

４６．請求項４１又は４２に記載のデータ変換装置において、

上記G関数手段は次の演算処理、

$(L_{j+1}, (Y_{j+1}, v_{j+1}))=G(Y_j, v_j)$（$0 \leqq j \leqq M-1$）において、出力結果を

$((Y_j^{(1)}, Y_j^{(2)}, Y_j^{(3)}, v_j) \rightarrow$

$$\left((L_{j+1}^{(1)}, L_{j+1}^{(2)}, L_{j+1}^{(3)}, L_{j+1}^{(4)}), [(Y_{j+1}^{(1)}, Y_{j+1}^{(2)}, Y_{j+1}^{(3)}, Y_{j+1}^{(4)}), v_{j+1}]\right)$$

ここで、

$$Y_{j+1}^{(i)} = f(Y_j^{(i)}) \quad (i=1, 2, 3, 4)$$

$$L_{j+1}^{(0)} = v_j$$

$$L_{j+1}^{(i)} = f(L_{j+1}^{(i-1)}) \oplus Y_{j+1}^{(i)} \quad (i=1, 2, 3, 4)$$

$$v_{j+1} = L_{j+1}^{(4)}$$

を実行する手段であり、

上記H関数手段は次の演算処理

$$k_i = H(i, L_1, L_2, \cdots, L_M)$$

において、

$$q_{4i+j} = L_{j+1}^{(i+1)} \quad (i=0,\ 1,\ 2,\ 3)$$

$$(t_i^{(0)}, t_i^{(1)}, \ldots, t_i^{(7)}) = q_i \quad (i=0,\ 1,\ \cdots,\ 31)$$

$$k_{(i+1)} = \left(t_{0+(i \bmod 2)}^{([i/2])}, t_{2+(i \bmod 2)}^{([i/2])}, \ldots, t_{30+(i \bmod 2)}^{([i/2])}\right) \quad (i=0,\ 1,\ \cdots,\ N-1)$$

を実行する手段である。

４７．　主鍵から副鍵をスケジュールする装置であって、

主鍵Kが入力され、縦続的に動作するM段からなり、各段より中間値L_j（j＝0, 1, …, M-1）成分を生成するG関数手段と、

そのG関数手段の出力である各中間値L_j成分を一旦記憶しておくための中間値記憶手段と、

複数個のL_j成分からN個の副鍵を生成する部分情報抽出機能を備えたH関数手段、

とを含み、

上記G関数手段は、上記主鍵Kを、Y_0の少なくとも一部として取り込み、j+1段目が（j＝0, 1, …, M-1）j段目の出力（L_j, Y_j, v_j）中のY_jとv_jを入力し、その入力を拡散して、L_{j+1}, Y_{j+1}, v_{j+1}を出力し、をKとする手段であり、

上記H関数手段は、i（i＝1, 2, …, N）と上記中間値記憶手段のL_1, L_2, …, L_Mとを入力とし、iによって決められたビット位置の情報をL_1, …, L_Mより抽出する副鍵k_iを出力する手段であることを特徴とする暗号鍵スケジュール装置。

４８．　主鍵から副鍵をスケジュールする装置であって、

主鍵Kが入力され、縦続的に動作するM段からなり、各段より中間値L_j（j＝0, 1, …, M-1）成分を生成するG関数手段と、

そのG関数手段で生成された複数個のL_j成分から副鍵を生成する部分情報抽出機能を備えたH関数手段と、

そのH関数手段の出力を副鍵k_iに対応する値として記憶しておくための中間値記憶手段、

とを含み、

上記G関数手段は、上記主鍵Kを、Y_0 の少なくとも一部として取り込み、j+1段目がj段目の出力（L_j, Y_j, v_j）中のY_j, v_jを入力し、その入力を拡散してL_{j+1}, Y_{j+1}, v_{j+1} を出力する手段であり、

上記H関数手段は、i, q, L_j （$1 \leqq i \leqq N$, $1 \leqq j \leqq M$, $1 \leqq q \leqq k_i$ のビット数）を入力として、L_j から、iとqによって決められたビット位置情報を抽出して、副鍵k_i のビット位置qの情報とする手段であることを特徴とする暗号鍵スケジュール装置。

４９．請求項４７又は４８に記載の装置において、上記G関数手段は、

入力されたY_jを２つのブロック(Y_j^L, Y_j^R)に分割し、Y_j^Lをv_{j+1}として出力するデータ分割手段と、

上記Y_j^Rと上記v_jとから$Y_j^R \oplus v_j$を演算する排他的論理和手段と、

上記Y_j^L と上記排他的論理和手段の出力とがあたえられ、それらを互いに拡散した結果をL_{j+1}しとして出力するデータ拡散手段と、

上記Y_j^RをY_{j+1}^Lとし、上記L_{j+1}をY_{j+1}^Rとし、これらY_{j+1}^LとY_{j+1}^Rを互いに連結して$Y_{j+1}=(Y_{j+1}^L, Y_{j+1}^R)$として出力するデータ入れ換え手段、

とを含む。

５０．請求項４７に記載の装置において、上記H関数手段は、

上記中間値記憶手段から読み出した各L_jを各１ビットずつ

$$(t_j^{(1)}, t_j^{(2)}, \cdots, t_j^{(2N)}) = L_j \quad (j=1, 2, \cdots, M)$$

に分割するビット分割手段と、

得られた$(t_1^{(i)}, t_1^{(N+i)}, t_2^{(i)}, t_2^{(N+i)}, \cdots, t_M^{(i)}, t_M^{(N+i)})$を結合して副鍵

$$k_i = (t_1^{(i)}, t_1^{(N+i)}, t_2^{(i)}, t_2^{(N+i)}, \cdots, t_M^{(i)}, t_M^{(N+i)}) \quad (i=1, 2, \cdots, N)$$

を出力するビット結合手段、

とを含む。

５１．請求項４８に記載の装置において、上記H関数手段は、

上記各L_jを各１ビットずつ

$$(t_j^{(1)},\ t_j^{(2)},\ \cdots,\ t_j^{(2N)}) = L_j \quad (j=1, 2, \cdots, M)$$

に分割するビット分割手段と、

上記ビット $(t_j^{(1)},\ t_j^{(2)},\ \cdots,\ t_j^{(2N)})$ からiに対してk_iのビット位置qにより決まるLjのビット位置の情報が、k_iのビット位置となるよう結合して副鍵

$$k_i = (t_1^{(i)},\ t_1^{(N+i)},\ t_2^{(i)},\ t_2^{(N+i)},\ \cdots,\ t_M^{(i)},\ t_M^{(N+i)})$$
$$(i=1, 2, \cdots, N)$$

を出力するビット結合手段、

とを含む。

５２．請求項４７又は４８に記載の装置において、

上記Ｇ関数手段は次の演算処理、

$(L_{j+1},\ (Y_{j+1},\ v_{j+1})) = G(Y_j,\ v_j)$ （$0 \leqq j \leqq M-1$）において、出力結果を

$((Y_j^{(1)},\ Y_j^{(2)},\ Y_j^{(3)}),\ v_j) \rightarrow$

$$\left((L_{j+1}^{(1)}, L_{j+1}^{(2)}, L_{j+1}^{(3)}, L_{j+1}^{(4)}), [(Y_{j+1}^{(1)}, Y_{j+1}^{(2)}, Y_{j+1}^{(3)}, Y_{j+1}^{(4)}), v_{j+1}]\right)$$

ここで、

$$Y_{j+1}^{(i)} = f(Y_j^{(i)}) \quad (i=1,\ 2,\ 3,\ 4)$$

$$L_{j+1}^{(0)} = v_j$$

$$L_{j+1}^{(i)} = f(L_{j+1}^{(i-1)}) \oplus Y_{j+1}^{(i)} \quad (i=1,\ 2,\ 3.\ 4)$$

$$v_{j+1} = L_{j+1}^{(4)}$$

を実行する手段であり、

上記Ｈ関数手段は次の演算処理

$$k_i = H(i,\ L_1,\ L_2,\ \cdots,\ L_M)$$

において、

$$q_{4i+j} = L_{j+1}^{(i+1)} \quad (i=0,\ 1,\ 2,\ 3)$$

$$(t_i^{(0)}, t_i^{(1)}, \ldots, t_i^{(7)}) = q_i \quad (i=0,\ 1,\ \cdots,\ 31)$$

$$k_{(i+1)} = \left(t_{0+(i \bmod 2)}^{([i/2])}, t_{2+(i \bmod 2)}^{([i/2])}, \ldots, t_{30+(i \bmod 2)}^{([i/2])}\right) \quad (i=0,\ 1,\ \cdots,\ N-1)$$

を実行する手段である。

５３．　主鍵Ｋを入力して複数の副鍵k_i（ｉ＝１，…，Ｎ）を生成する暗号鍵スケジュール装置をコンピュータにより実行させるためのプログラムを記録した記録媒体であって、

上記プログラムは、Y_0としての主鍵Ｋと定数v_0とを入力として入力を拡散処理することを縦続的に複数回繰返し、各拡散処理ごとに中間鍵L_j（j＝1，2，…，M）を出力する中間鍵生成処理と、

上記各生成された中間鍵L_jを記憶部に記憶する処理と、

上記記憶部に所定数の中間鍵L_1～L_Mが記憶されると、副鍵k_iのｉによって決まる各ビット位置のL_1～L_Mにおける情報を抽出して副鍵k_iを生成する副鍵生成処理と、

を上記コンピュータにより実行させることを特徴とする記録媒体。

５４．　主鍵Ｋを入力して複数の副鍵k_i（i＝1，2，…，N）を生成する暗号鍵スケジュール装置をコンピュータにより実行させるためのプログラムを記録した記録媒体であって、

上記プログラムは、

Y_0としての主鍵Ｋと定数v_0とを入力として、入力を拡散処理することを縦続的に複数回繰返し、各拡散処理ごとに中間鍵L_j（j＝1，2，…，M）を出力する中間鍵生成処理と、

上記各中間鍵L_jが生成されるごとに、副鍵k_iのｉと、k_iのビット位置ｑとにより決まるL_jのビット位置の情報をk_iのビット位置の情報として抽出して、記憶部に記憶する処理と、

上記記憶部内の各副鍵k_iの各ビット位置の情報が決まると、その副鍵k_iを出力する処理と、

を処理コンピュータにより実行させることを特徴とする記録媒体。

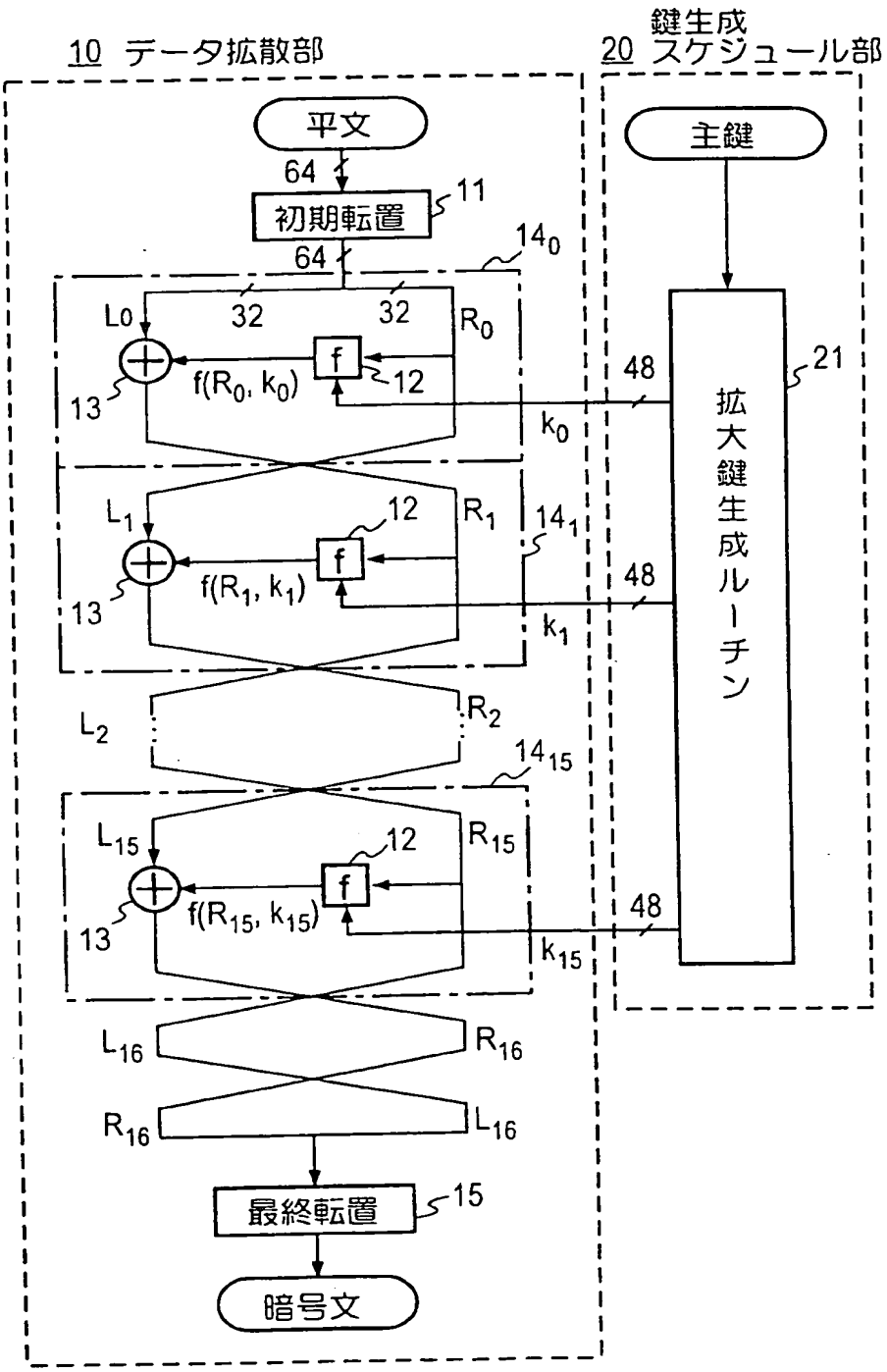

10 データ拡散部
鍵生成
20 スケジュール部
平文
64
初期転置
11
64
14_0
32
32
L_0
R_0
f
12
13
$f(R_0, k_0)$
k_0
48
主鍵
21
拡大鍵生成ルーチン
L_1
R_1
14_1
$f(R_1, k_1)$
k_1
L_2
R_2
14_{15}
L_{15}
R_{15}
$f(R_{15}, k_{15})$
k_{15}
L_{16}
R_{16}
R_{16}
L_{16}
最終転置
15
暗号文

図1

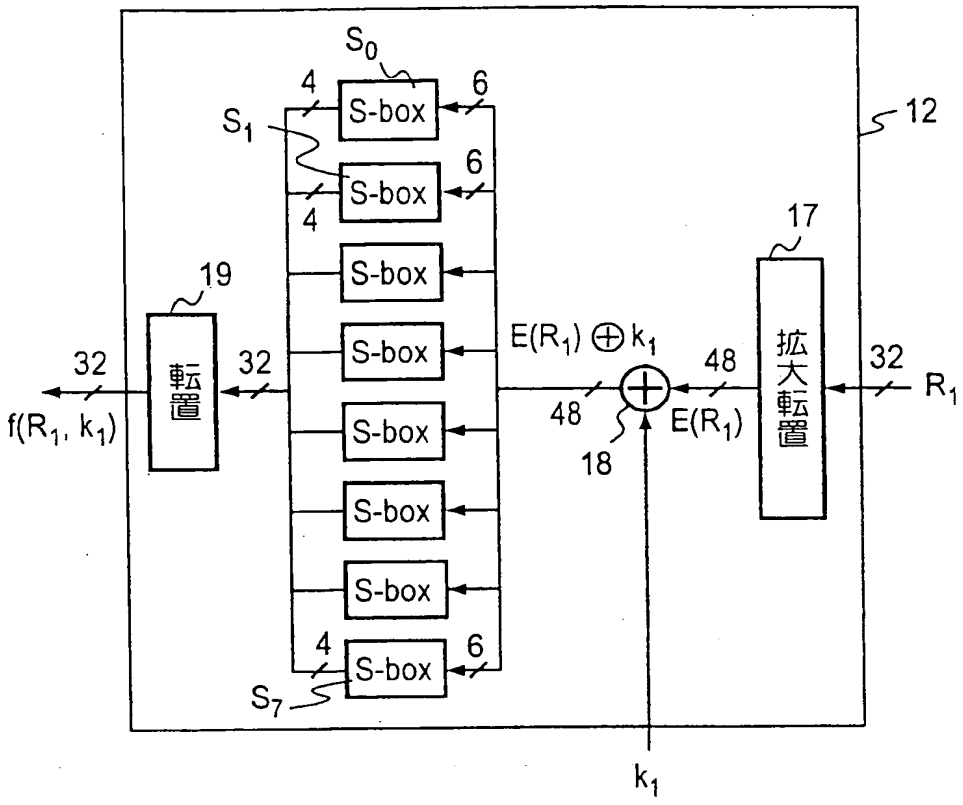

S0
S1
S7
S-box
4
6
12
17
19
32
32
48
48
18
E(R1) ⊕ k1
E(R1)
f(R1, k1)
R1
転置
拡大転置
k1

図2

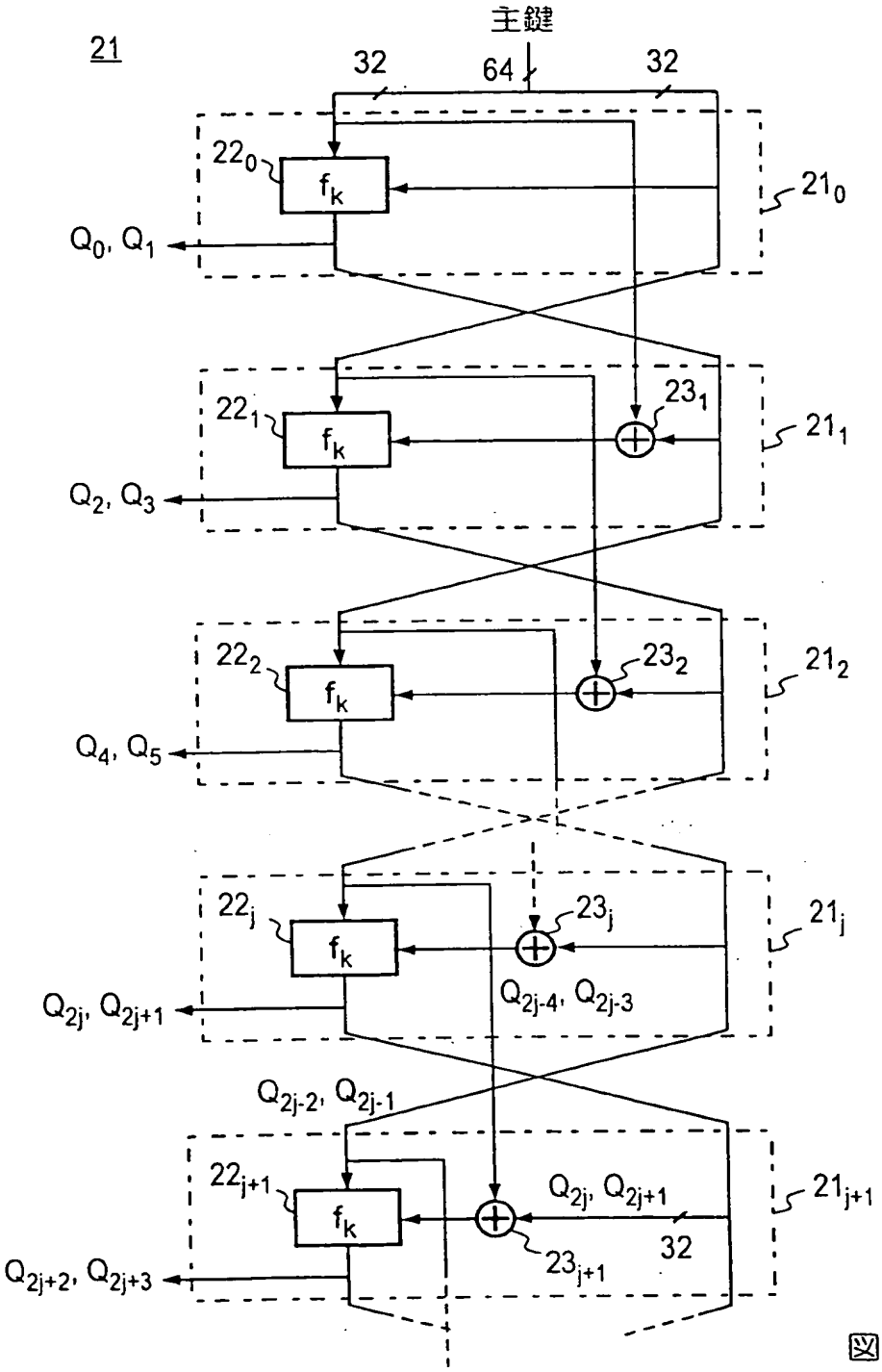

主鍵
21
32
64
32
22_0
f_k
21_0
Q_0, Q_1
22_1
23_1
21_1
Q_2, Q_3
22_2
23_2
21_2
Q_4, Q_5
22_j
23_j
21_j
Q_{2j}, Q_{2j+1}
Q_{2j-4}, Q_{2j-3}
Q_{2j-2}, Q_{2j-1}
22_{j+1}
Q_{2j}, Q_{2j+1}
21_{j+1}
32
23_{j+1}
Q_{2j+2}, Q_{2j+3}

図3

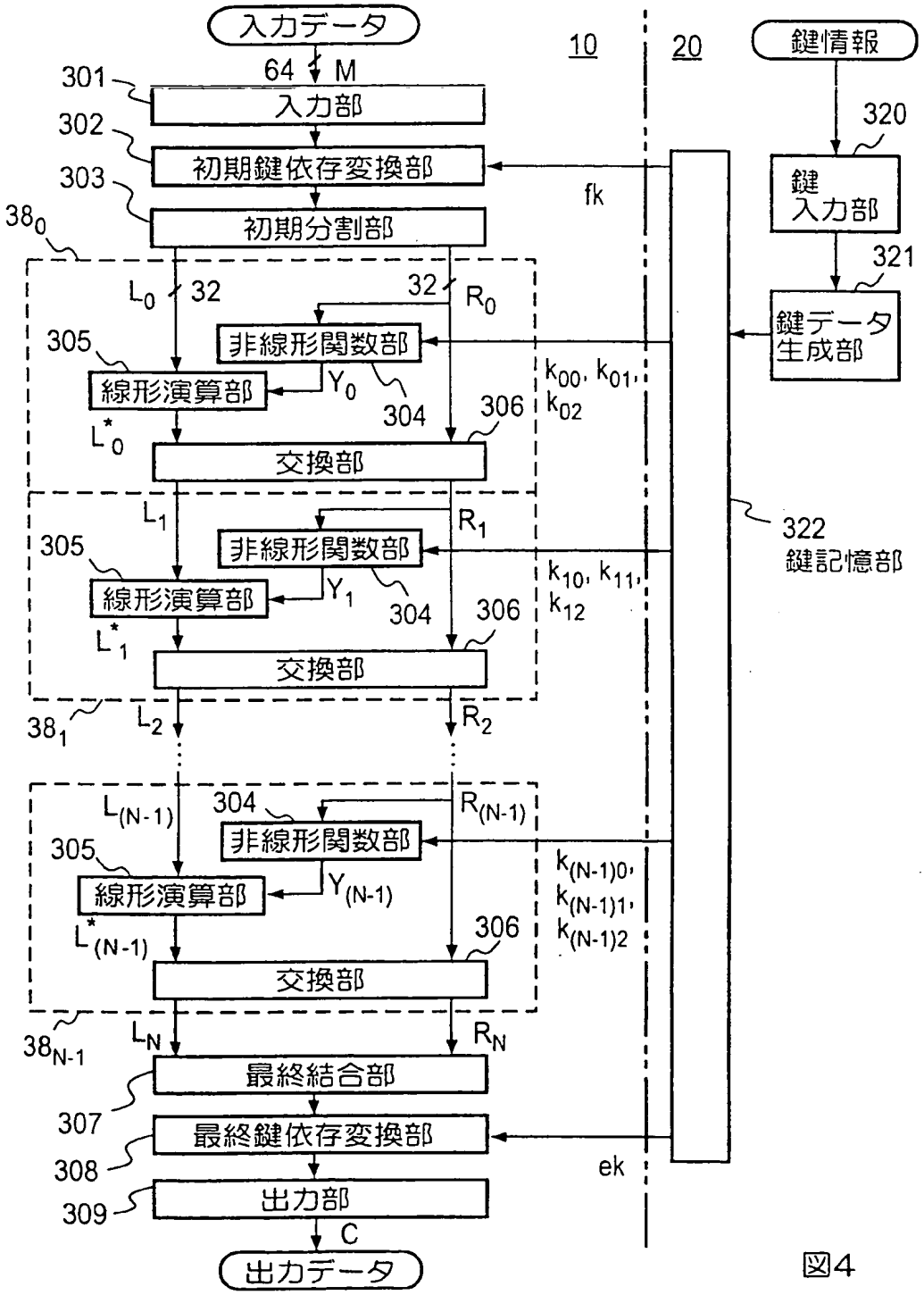

入力データ
64
M
10
20
鍵情報
301
入力部
302
初期鍵依存変換部
303
初期分割部
320
鍵
入力部
321
鍵データ
生成部
fk
38₀
L₀
32
32
R₀
305
非線形関数部
線形演算部
Y₀
304
306
k₀₀, k₀₁,
k₀₂
L*₀
交換部
L₁
R₁
非線形関数部
線形演算部
Y₁
k₁₀, k₁₁,
k₁₂
L*₁
交換部
322
鍵記憶部
38₁
L₂
R₂
L(N-1)
R(N-1)
非線形関数部
線形演算部
Y(N-1)
k(N-1)0,
k(N-1)1,
k(N-1)2
L*(N-1)
交換部
38N-1
LN
RN
最終結合部
307
最終鍵依存変換部
308
ek
出力部
309
C
出力データ

図4

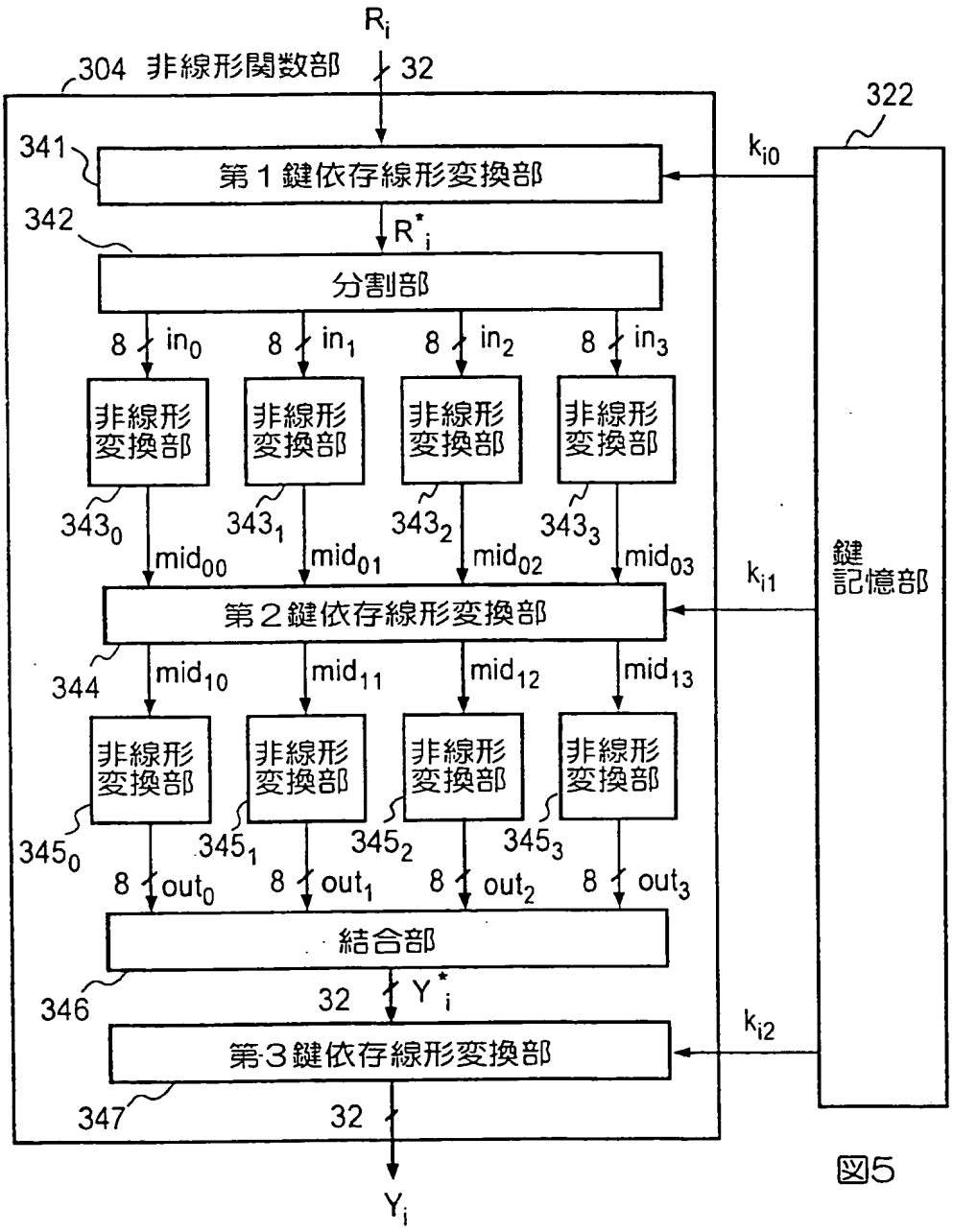

R_i
304 非線形関数部
32
322
341
第１鍵依存線形変換部
k_{i0}
342
R^*_i
分割部
8 in_0
8 in_1
8 in_2
8 in_3
非線形変換部
非線形変換部
非線形変換部
非線形変換部
343_0
343_1
343_2
343_3
mid_{00}
mid_{01}
mid_{02}
mid_{03}
k_{i1}
鍵記憶部
第２鍵依存線形変換部
344
mid_{10}
mid_{11}
mid_{12}
mid_{13}
非線形変換部
非線形変換部
非線形変換部
非線形変換部
345_0
345_1
345_2
345_3
8 out_0
8 out_1
8 out_2
8 out_3
結合部
346
32 Y^*_i
k_{i2}
第３鍵依存線形変換部
347
32
Y_i

図5

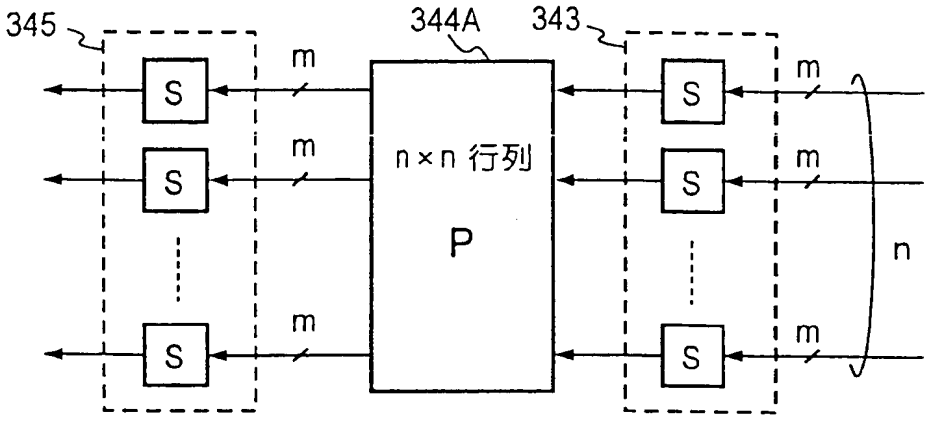

345
344A
343
S
m
n×n 行列
P
n

図6

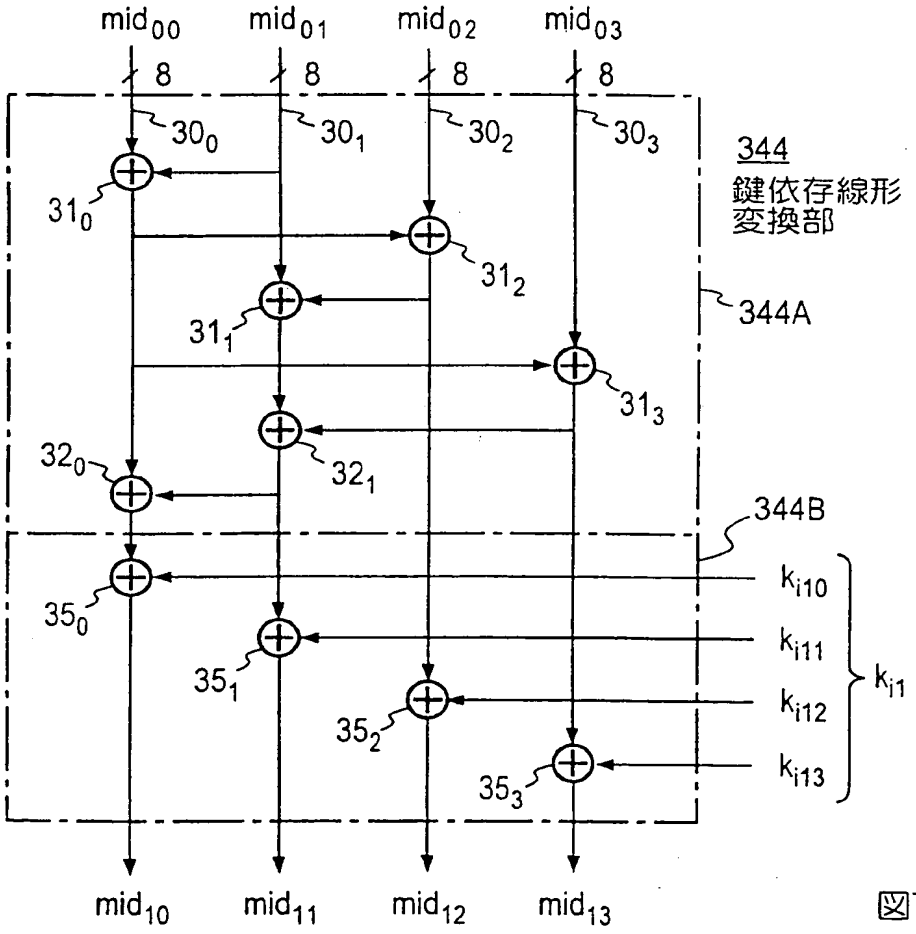

mid00
mid01
mid02
mid03
8
30
31
32
35
344
鍵依存線形
変換部
344A
344B
ki10
ki11
ki12
ki13
ki1
mid10
mid11
mid12
mid13

図7

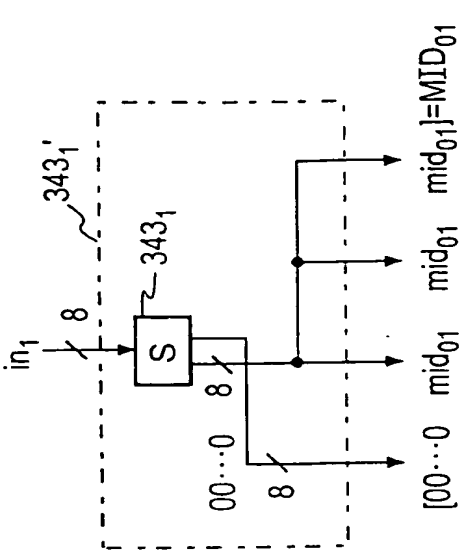


図8B

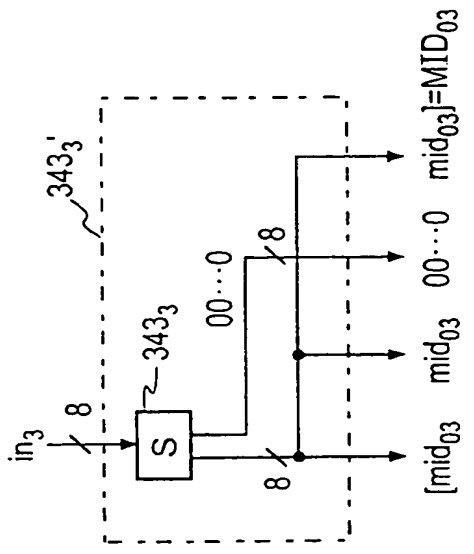


図8D

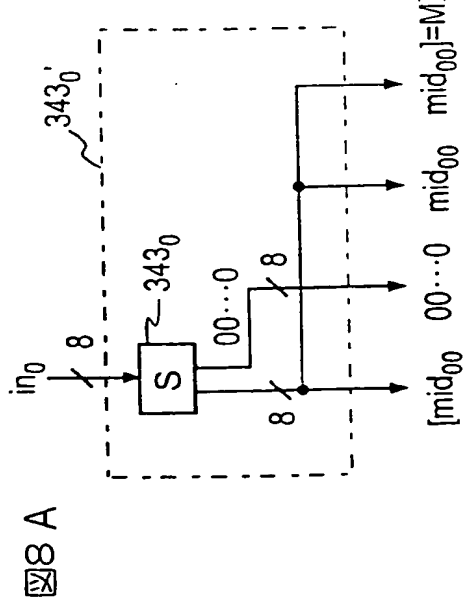


図8A

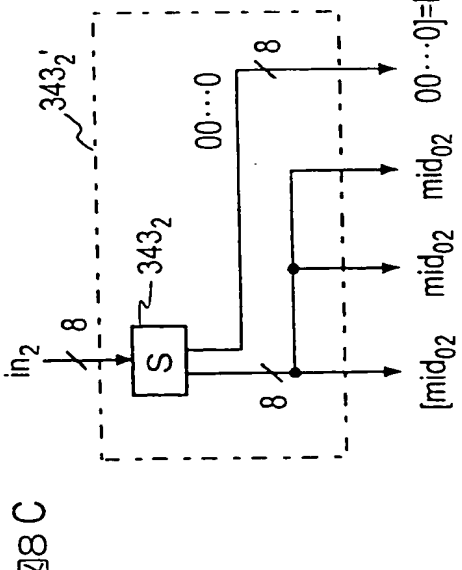


図8C

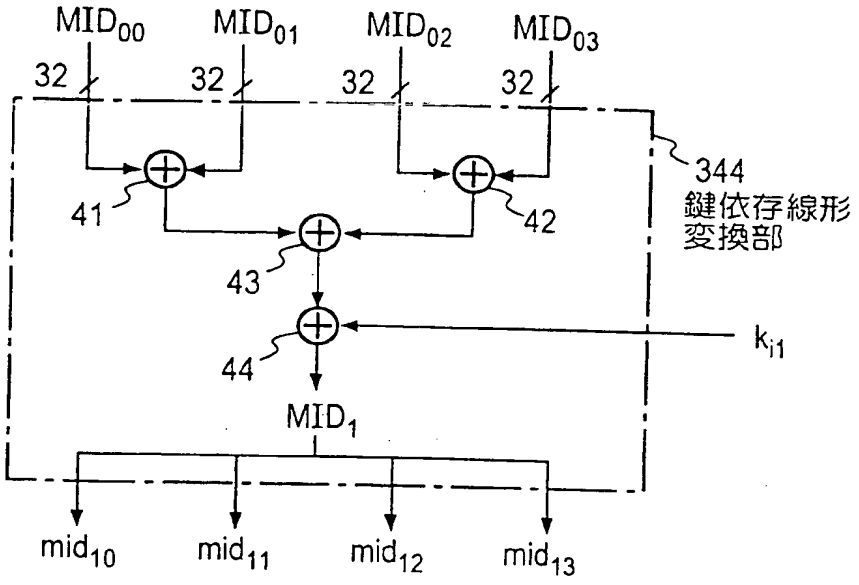

MID00
MID01
MID02
MID03
32
32
32
32
41
42
43
44
344
鍵依存線形
変換部
ki1
MID1
mid10
mid11
mid12
mid13

図9

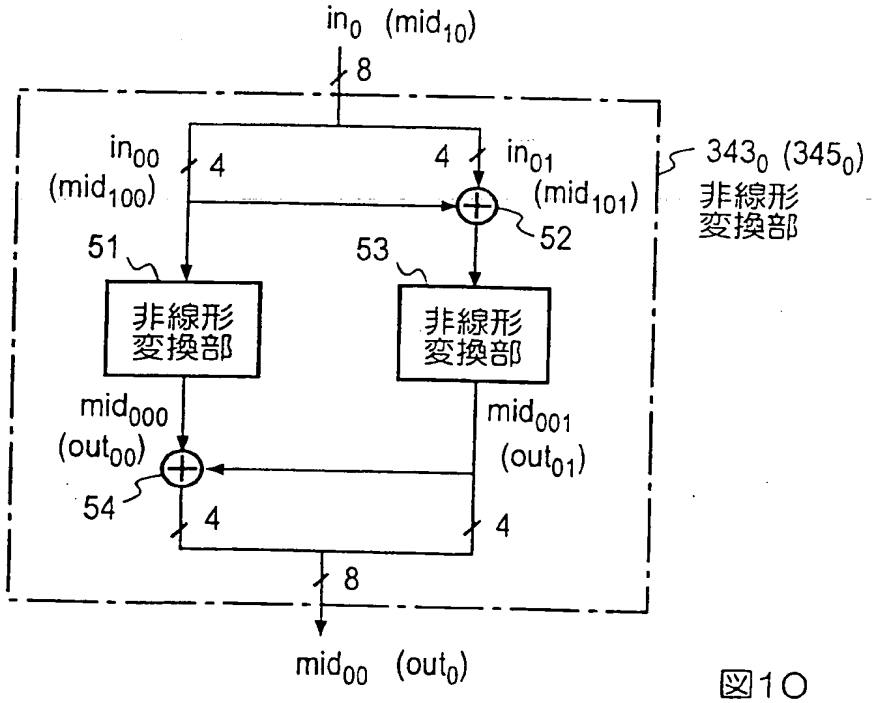

in0 (mid10)
8
in00
(mid100)
4
4
in01
(mid101)
343₀ (345₀)
非線形
変換部
51
52
53
非線形
変換部
非線形
変換部
mid000
(out00)
mid001
(out01)
54
4
4
8
mid00 (out0)

図10

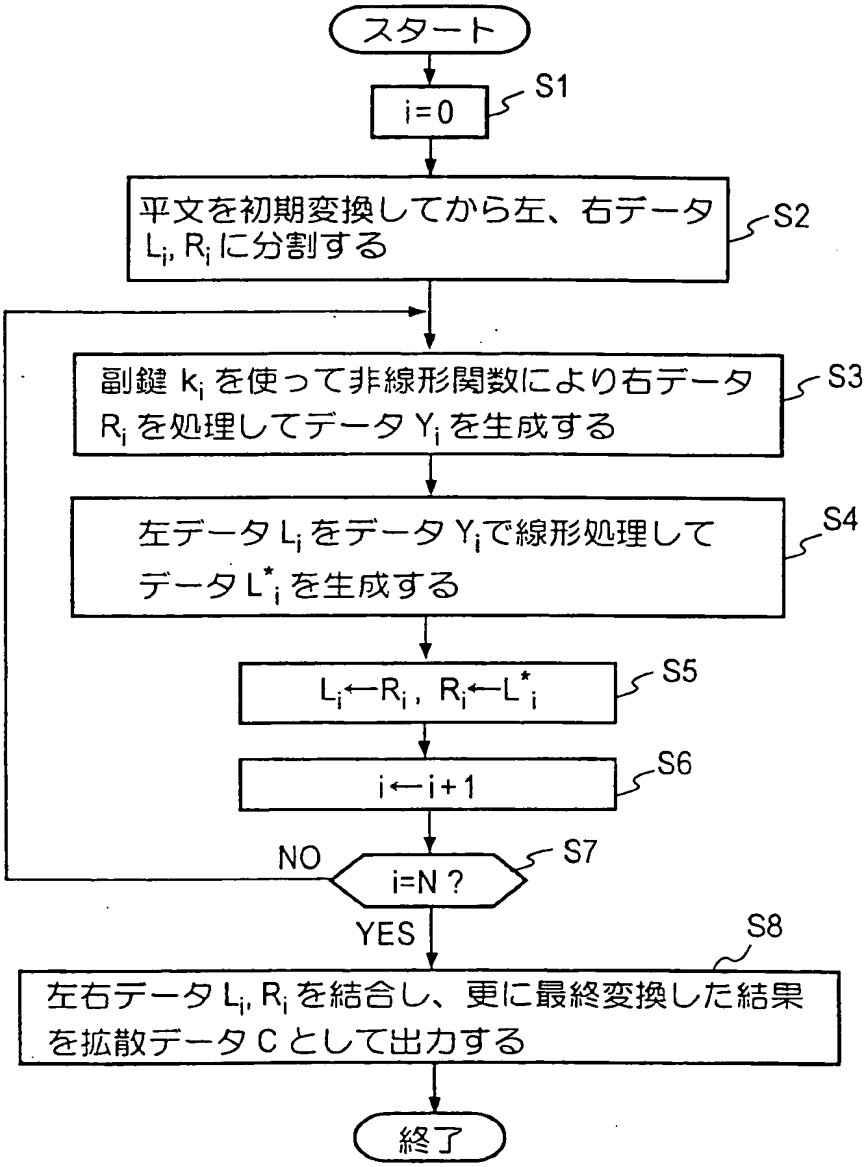

スタート
i=0
S1
平文を初期変換してから左、右データ L_i, R_i に分割する
S2
副鍵 k_i を使って非線形関数により右データ R_i を処理してデータ Y_i を生成する
S3
左データ L_i をデータ Y_i で線形処理してデータ L^*_i を生成する
S4
$L_i \leftarrow R_i$, $R_i \leftarrow L^*_i$
S5
$i \leftarrow i+1$
S6
NO
i=N ?
S7
YES
左右データ L_i, R_i を結合し、更に最終変換した結果を拡散データ C として出力する
S8
終了

図11

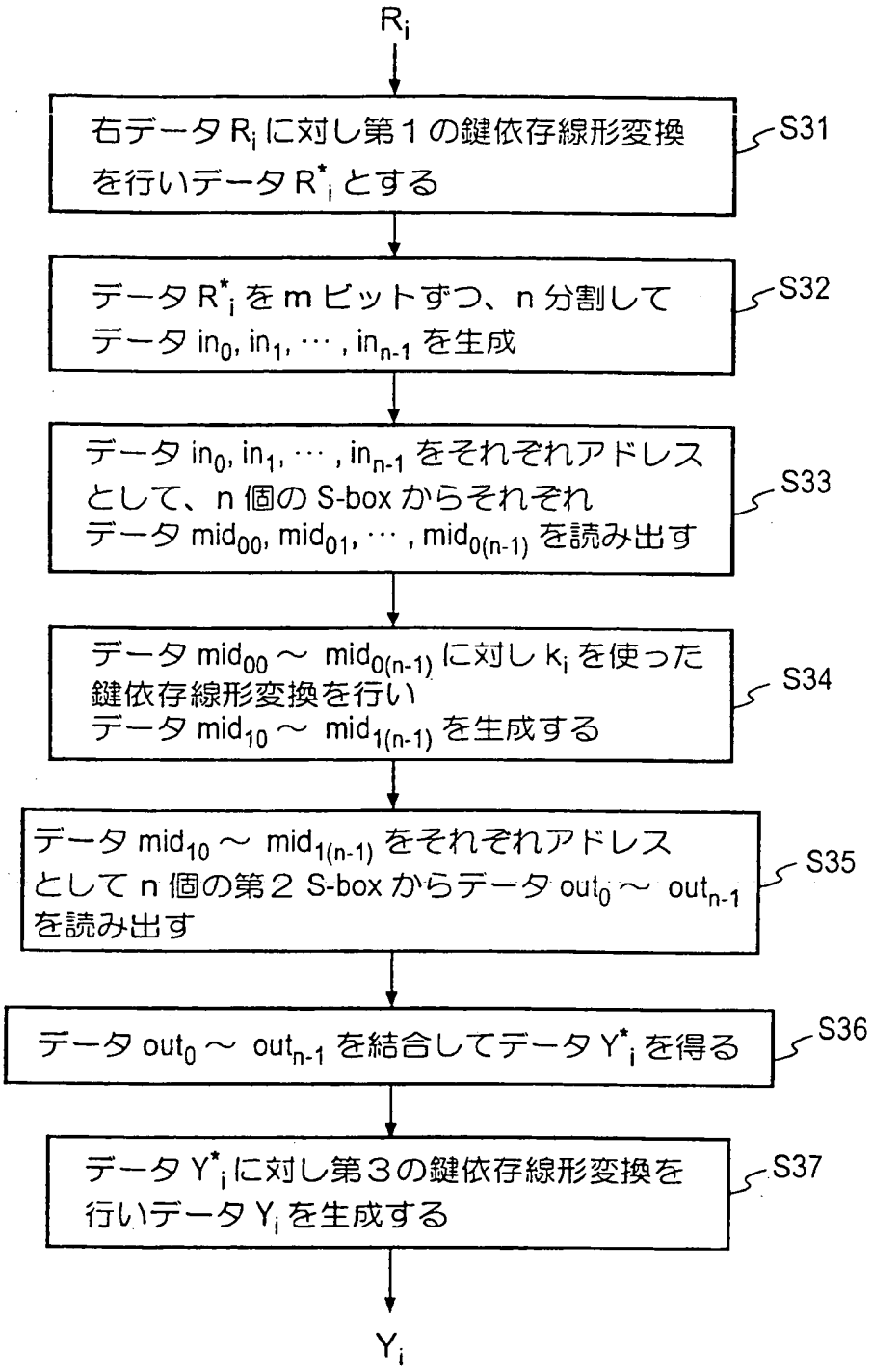

R_i
右データ R_i に対し第１の鍵依存線形変換を行いデータ R^*_i とする
S31
データ R^*_i を m ビットずつ、n 分割してデータ $in_0, in_1, \cdots, in_{n-1}$ を生成
S32
データ $in_0, in_1, \cdots, in_{n-1}$ をそれぞれアドレスとして、n 個の S-box からそれぞれデータ $mid_{00}, mid_{01}, \cdots, mid_{0(n-1)}$ を読み出す
S33
データ mid_{00} ～ $mid_{0(n-1)}$ に対し k_i を使った鍵依存線形変換を行いデータ mid_{10} ～ $mid_{1(n-1)}$ を生成する
S34
データ mid_{10} ～ $mid_{1(n-1)}$ をそれぞれアドレスとして n 個の第２ S-box からデータ out_0 ～ out_{n-1} を読み出す
S35
データ out_0 ～ out_{n-1} を結合してデータ Y^*_i を得る
S36
データ Y^*_i に対し第３の鍵依存線形変換を行いデータ Y_i を生成する
S37
Y_i

図12

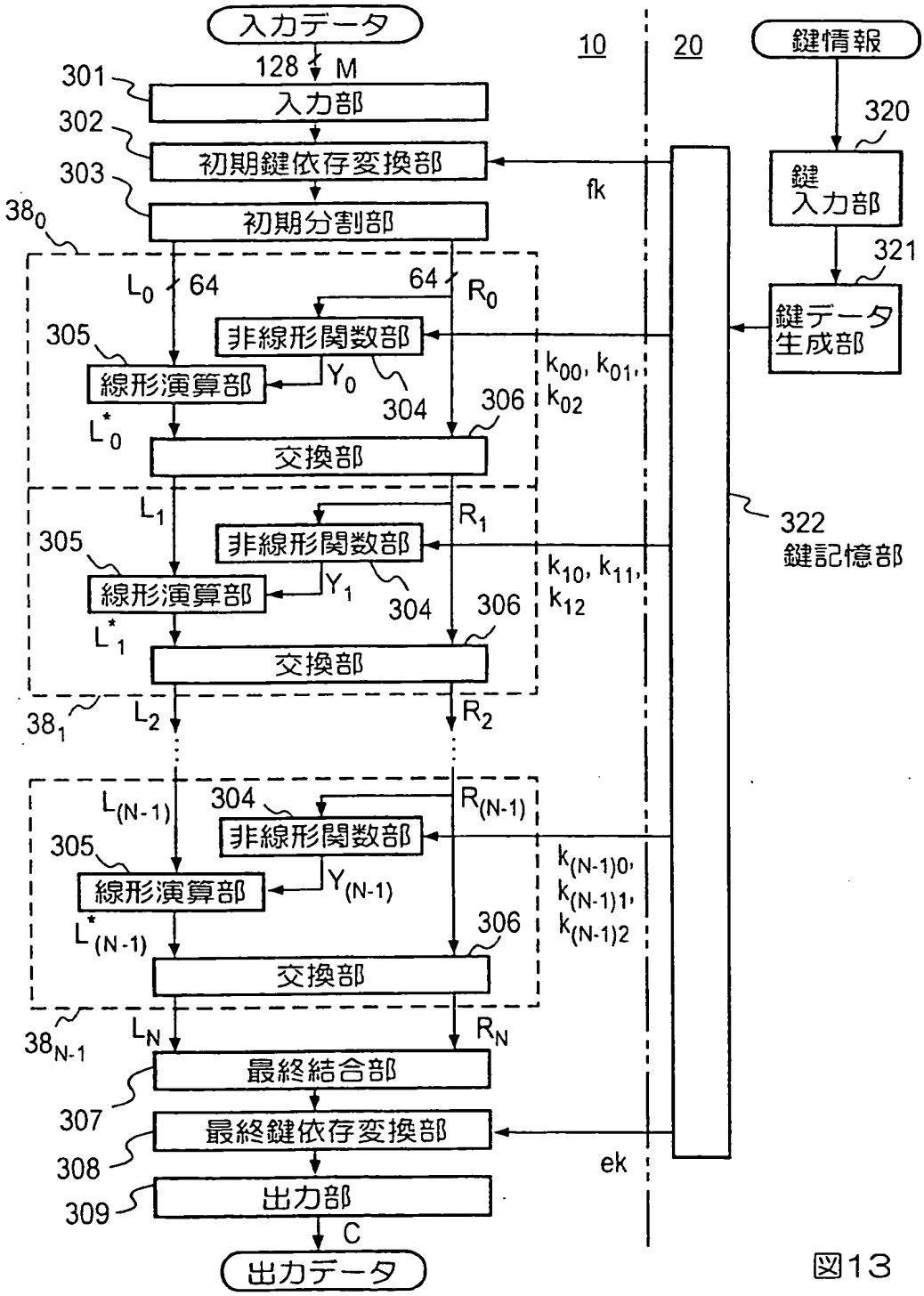

入力データ
128
M
10
20
鍵情報
301
入力部
302
初期鍵依存変換部
303
初期分割部
320
鍵
入力部
321
鍵データ
生成部
fk
38_0
L_0
64
64
R_0
305
非線形関数部
線形演算部
Y_0
304
306
k_{00}, k_{01},
k_{02}
L^*_0
交換部
L_1
R_1
k_{10}, k_{11},
k_{12}
L^*_1
Y_1
38_1
L_2
R_2
$L_{(N-1)}$
$R_{(N-1)}$
$k_{(N-1)0}$,
$k_{(N-1)1}$,
$k_{(N-1)2}$
$L^*_{(N-1)}$
$Y_{(N-1)}$
38_{N-1}
L_N
R_N
最終結合部
307
最終鍵依存変換部
308
ek
出力部
309
C
出力データ
322
鍵記憶部

図13

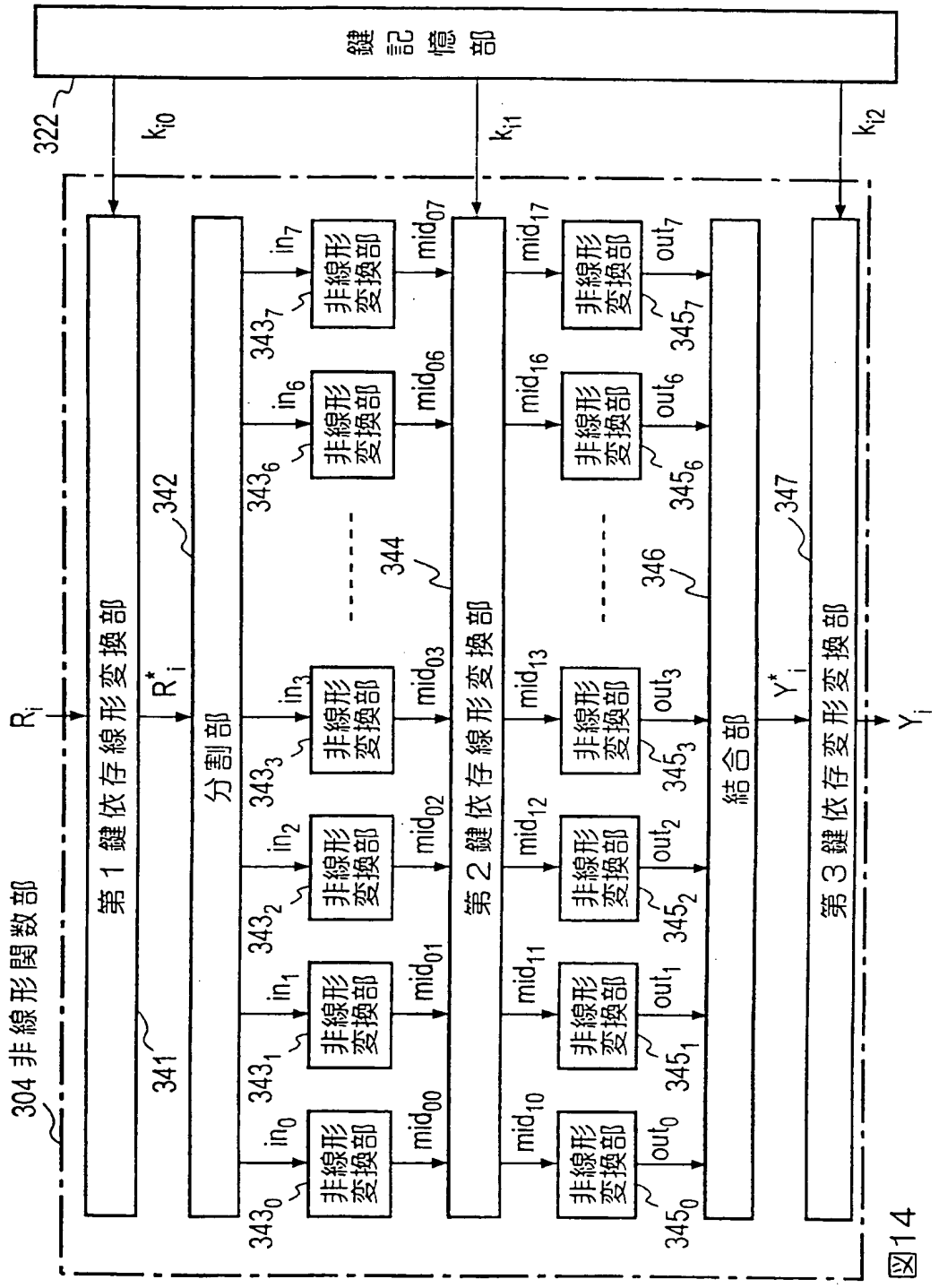

鍵記憶部
322
k_{i0}
k_{i1}
k_{i2}
304 非線形関数部
R_i
Y_i
第1鍵依存線形変換部
分割部
第2鍵依存線形変換部
結合部
第3鍵依存変形変換部
341
342
344
346
347
R^*_i
Y^*_i
343_0
343_1
343_2
343_3
343_6
343_7
345_0
345_1
345_2
345_3
345_6
345_7
in_0
in_1
in_2
in_3
in_6
in_7
out_0
out_1
out_2
out_3
out_6
out_7
非線形変換部
mid_{00}
mid_{01}
mid_{02}
mid_{03}
mid_{06}
mid_{07}
mid_{10}
mid_{11}
mid_{12}
mid_{13}
mid_{16}
mid_{17}

図14

図15A

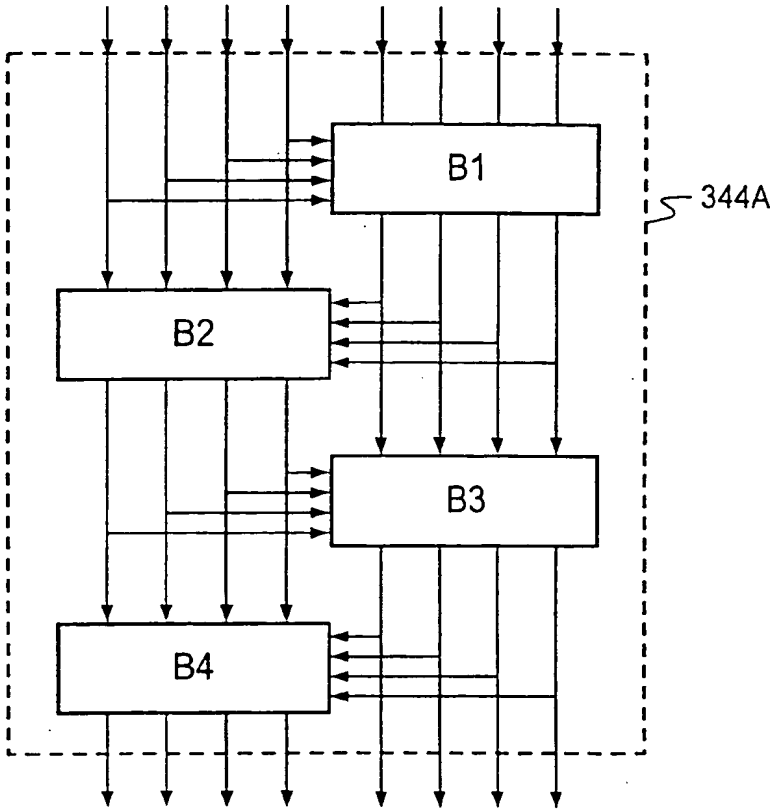
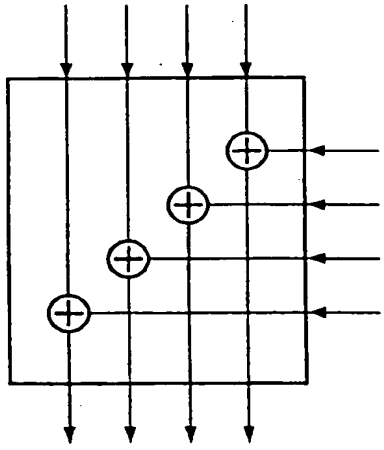


図15B



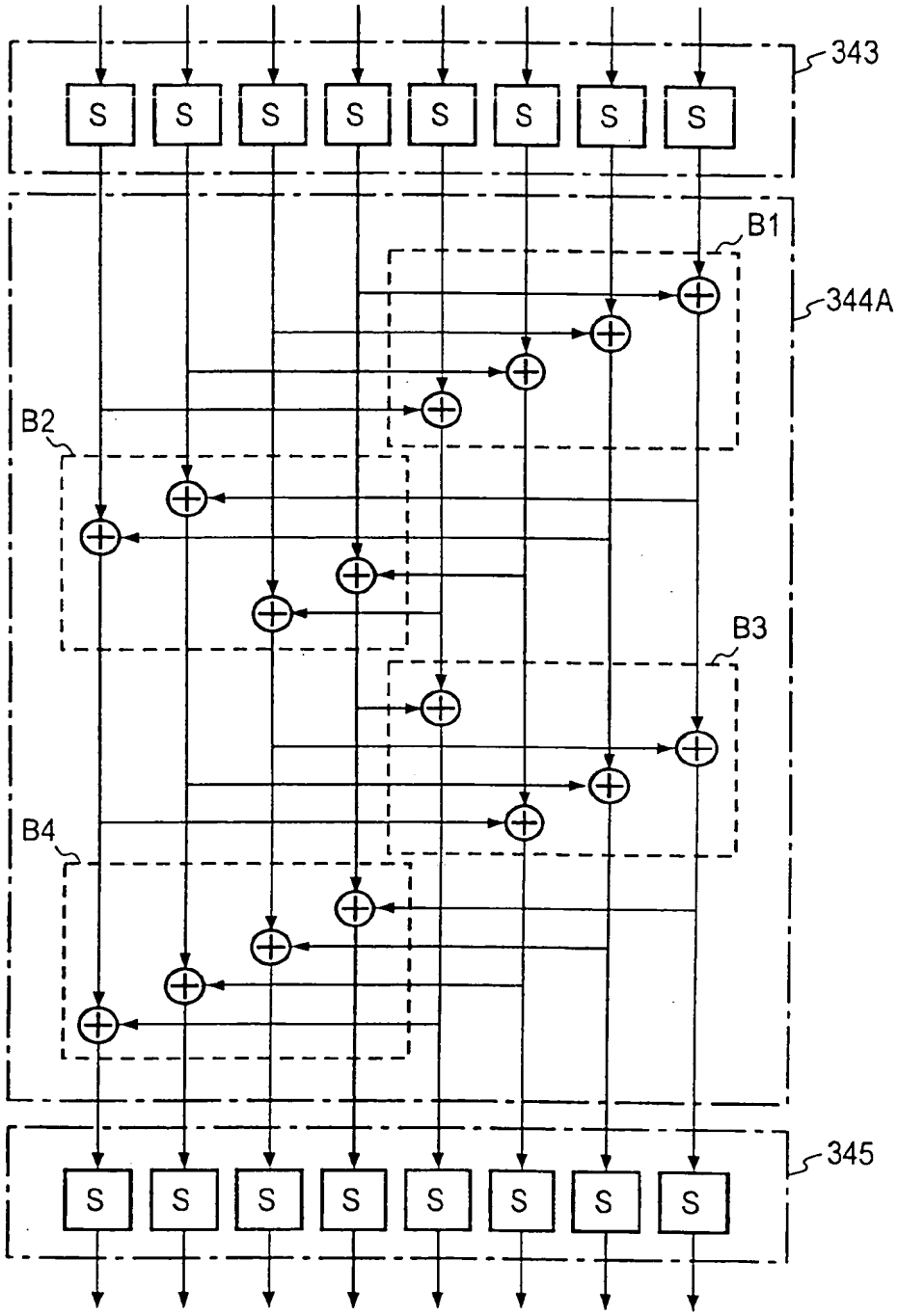


図16

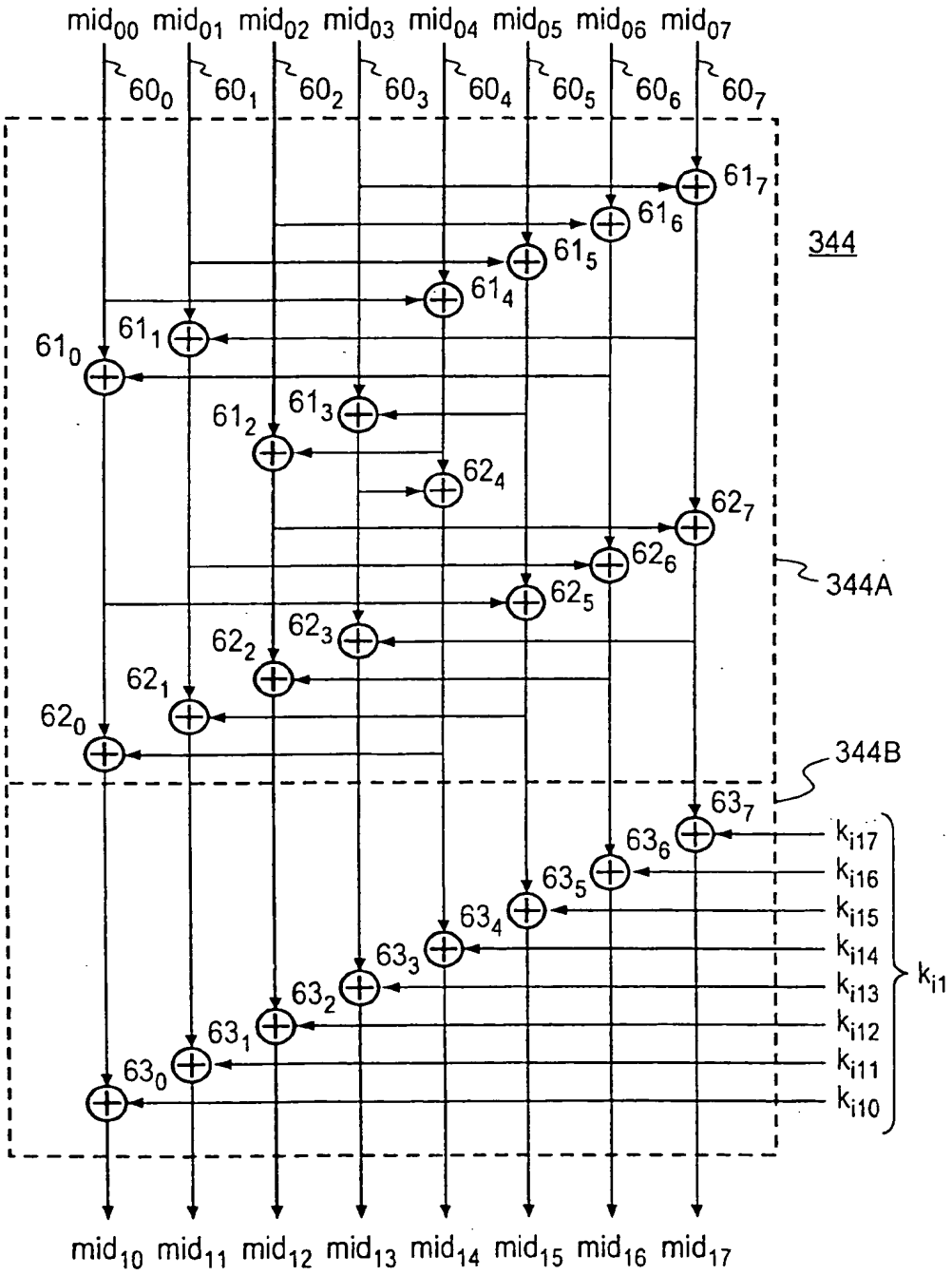

mid00 mid01 mid02 mid03 mid04 mid05 mid06 mid07
600 601 602 603 604 605 606 607
617 616 615 614 611 610 613 612 624 627 626 625 623 622 621 620
344
344A
344B
637 636 635 634 633 632 631 630
ki17 ki16 ki15 ki14 ki13 ki12 ki11 ki10
ki1
mid10 mid11 mid12 mid13 mid14 mid15 mid16 mid17

図17

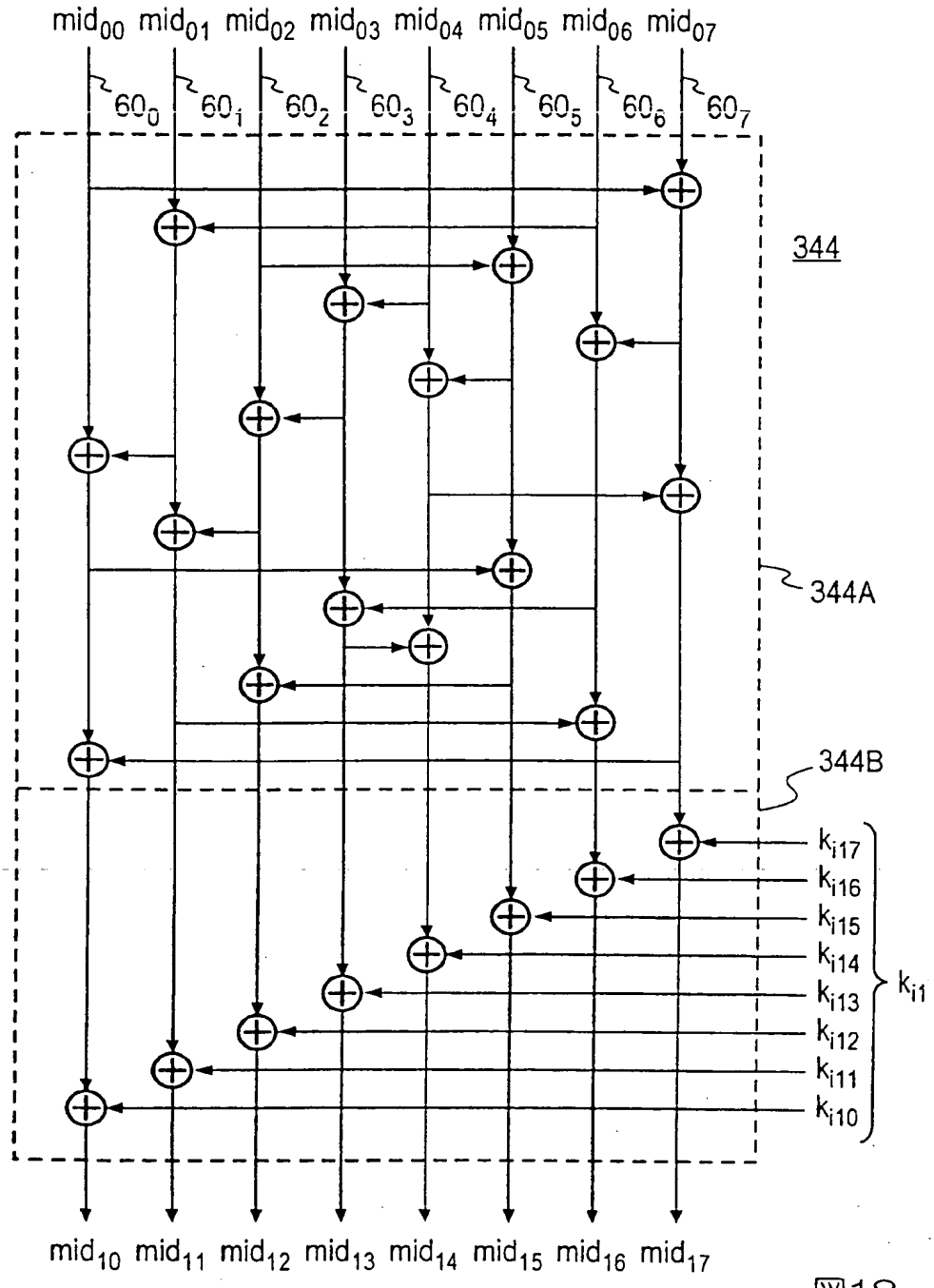

mid00 mid01 mid02 mid03 mid04 mid05 mid06 mid07
600 601 602 603 604 605 606 607
344
344A
344B
ki17
ki16
ki15
ki14
ki13
ki12
ki11
ki10
ki1
mid10 mid11 mid12 mid13 mid14 mid15 mid16 mid17

図18

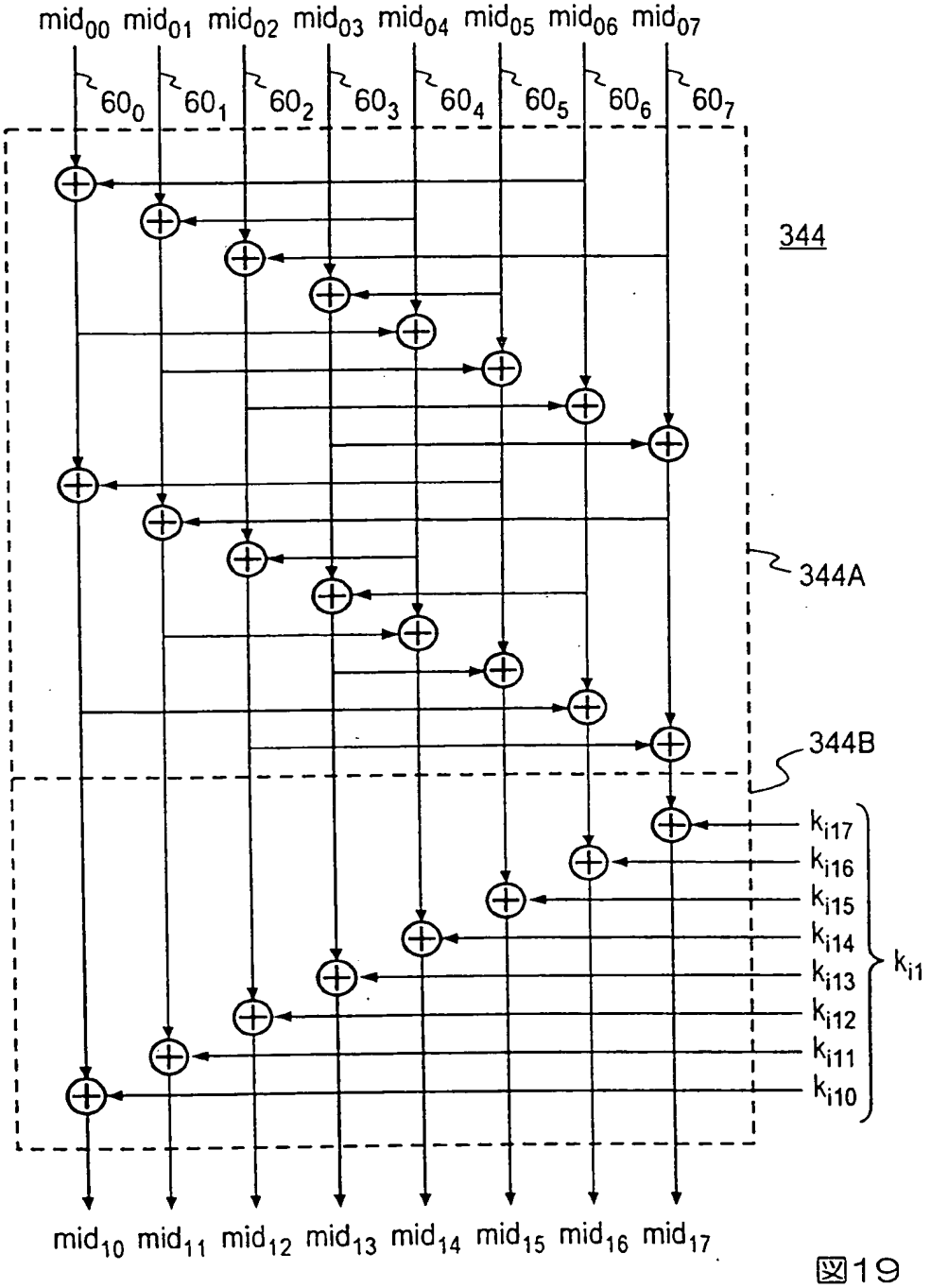

mid00 mid01 mid02 mid03 mid04 mid05 mid06 mid07
600 601 602 603 604 605 606 607
344
344A
344B
ki17
ki16
ki15
ki14
ki13
ki12
ki11
ki10
ki1
mid10 mid11 mid12 mid13 mid14 mid15 mid16 mid17

図19

図20 A

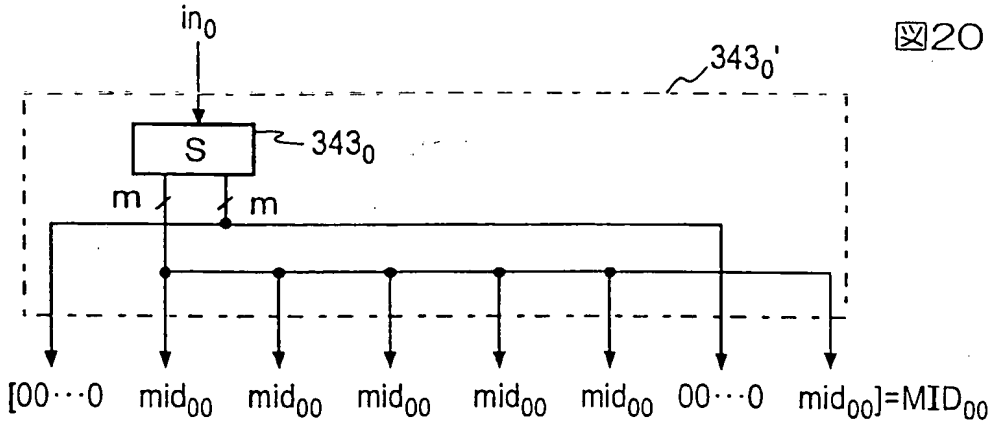


図20 B

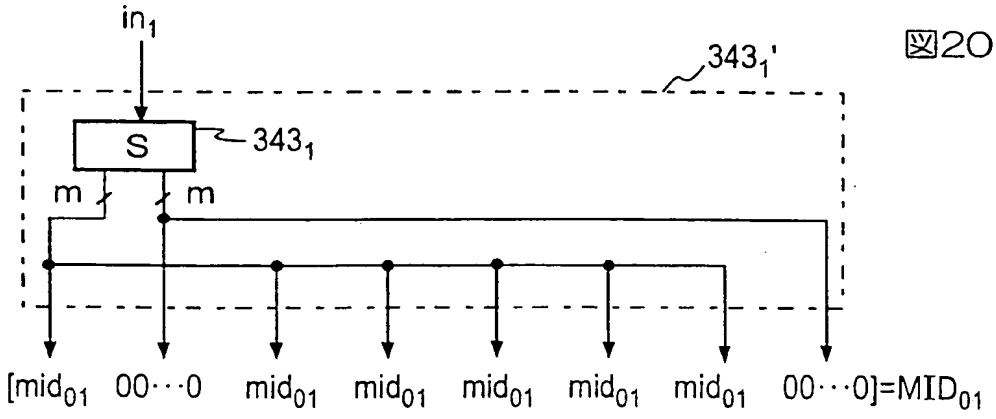
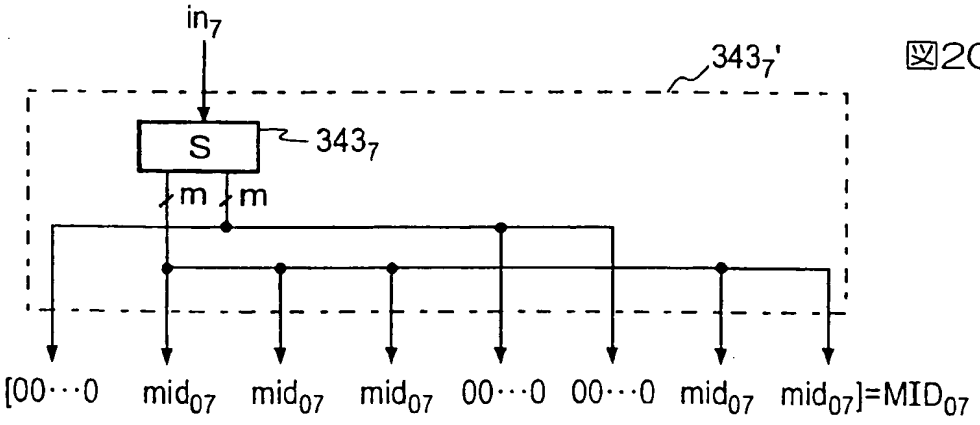


図20 C



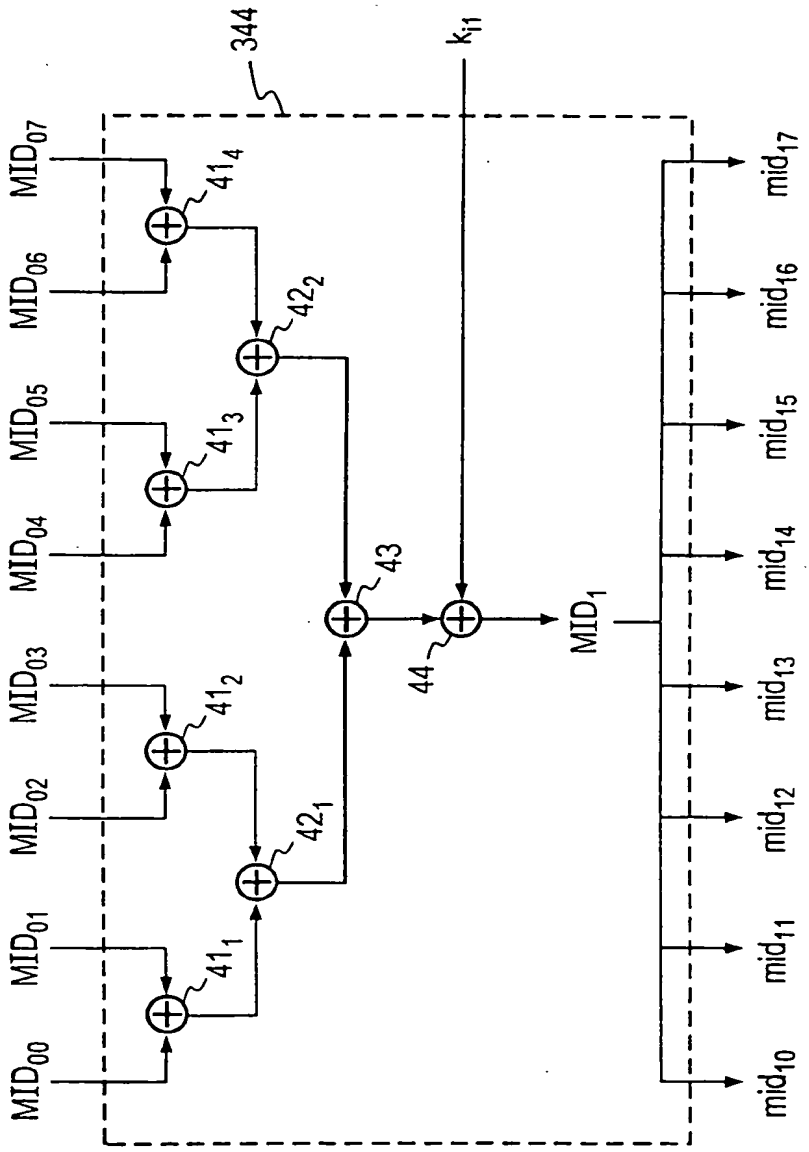

344
k_{11}
MID_{07}
MID_{06}
MID_{05}
MID_{04}
MID_{03}
MID_{02}
MID_{01}
MID_{00}
41_4
42_2
41_3
43
41_2
42_1
41_1
44
MID_1
mid_{17}
mid_{16}
mid_{15}
mid_{14}
mid_{13}
mid_{12}
mid_{11}
mid_{10}

図21

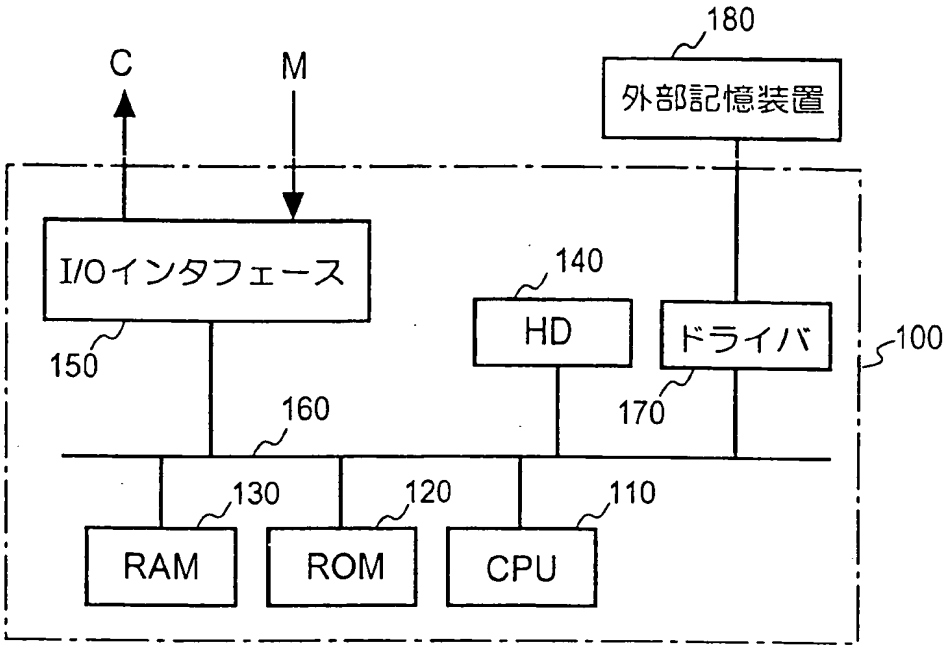

C
M
180
外部記憶装置
I/Oインタフェース
140
HD
ドライバ
100
150
160
170
130
120
110
RAM
ROM
CPU

図22

図23B

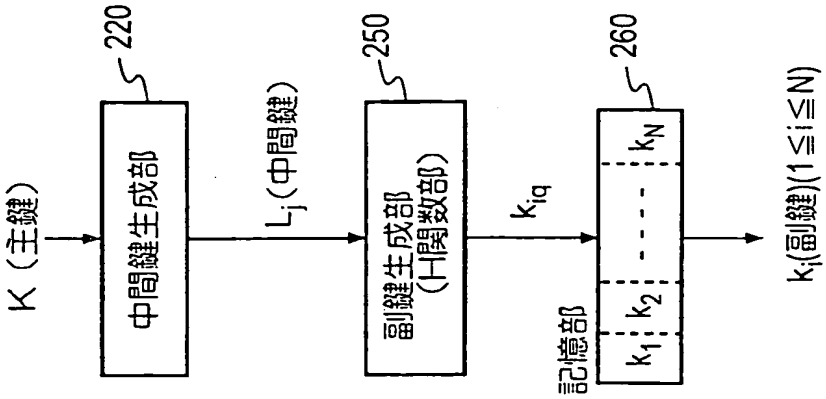
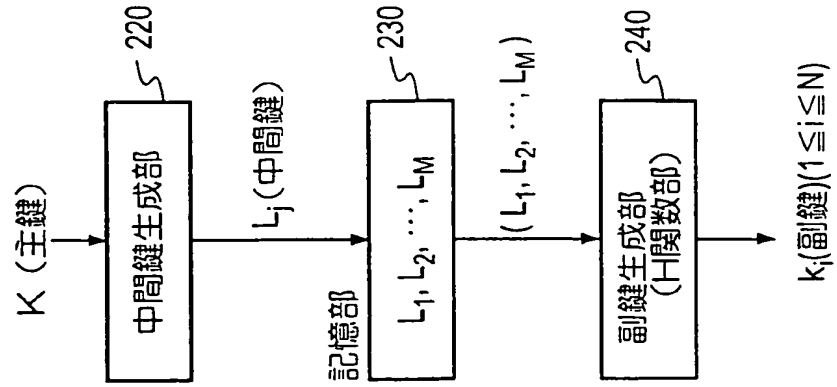

K（主鍵）
中間鍵生成部
220
L_j（中間鍵）
副鍵生成部
(H関数部)
250
k_{iq}
記憶部
k_1 k_2 - - - - k_N
260
k_i(副鍵)(1≦i≦N)

図23A


K（主鍵）
中間鍵生成部
220
L_j（中間鍵）
記憶部
$L_1, L_2, \cdots, L_M$
230
$(L_1, L_2, \cdots, L_M)$
副鍵生成部
(H関数部)
240
k_i(副鍵)(1≦i≦N)

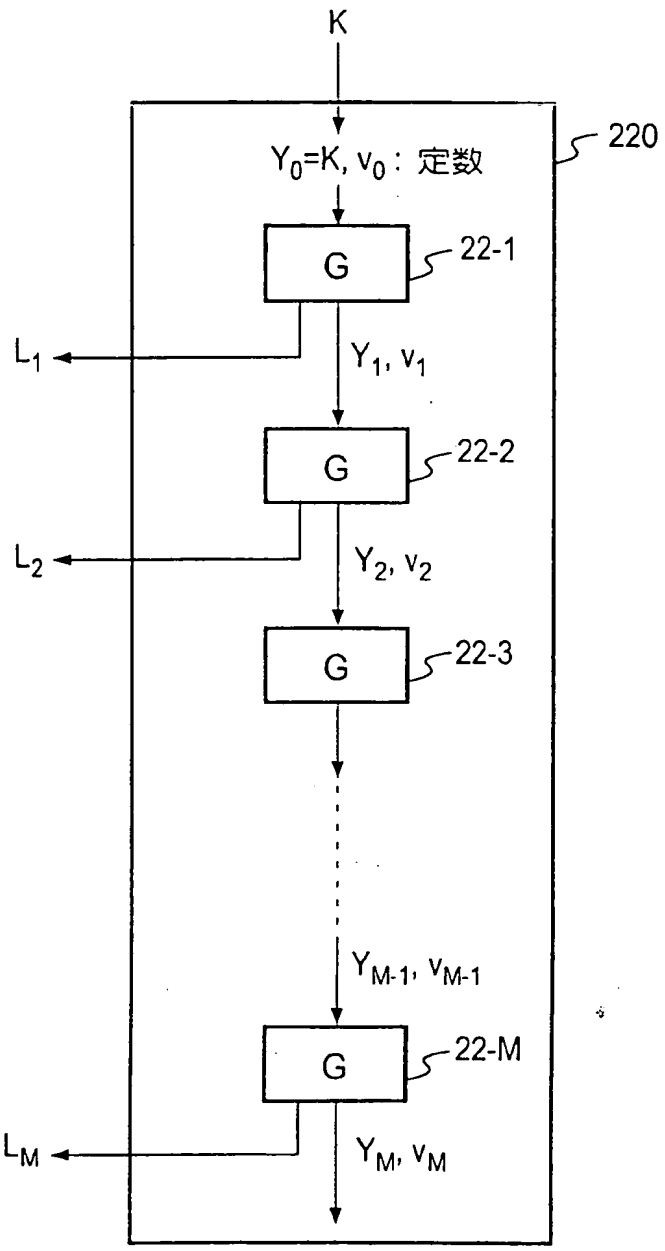

K
220
Y_0=K, v_0 : 定数
G
22-1
L_1
Y_1, v_1
G
22-2
L_2
Y_2, v_2
G
22-3
Y_{M-1}, v_{M-1}
G
22-M
L_M
Y_M, v_M

図24

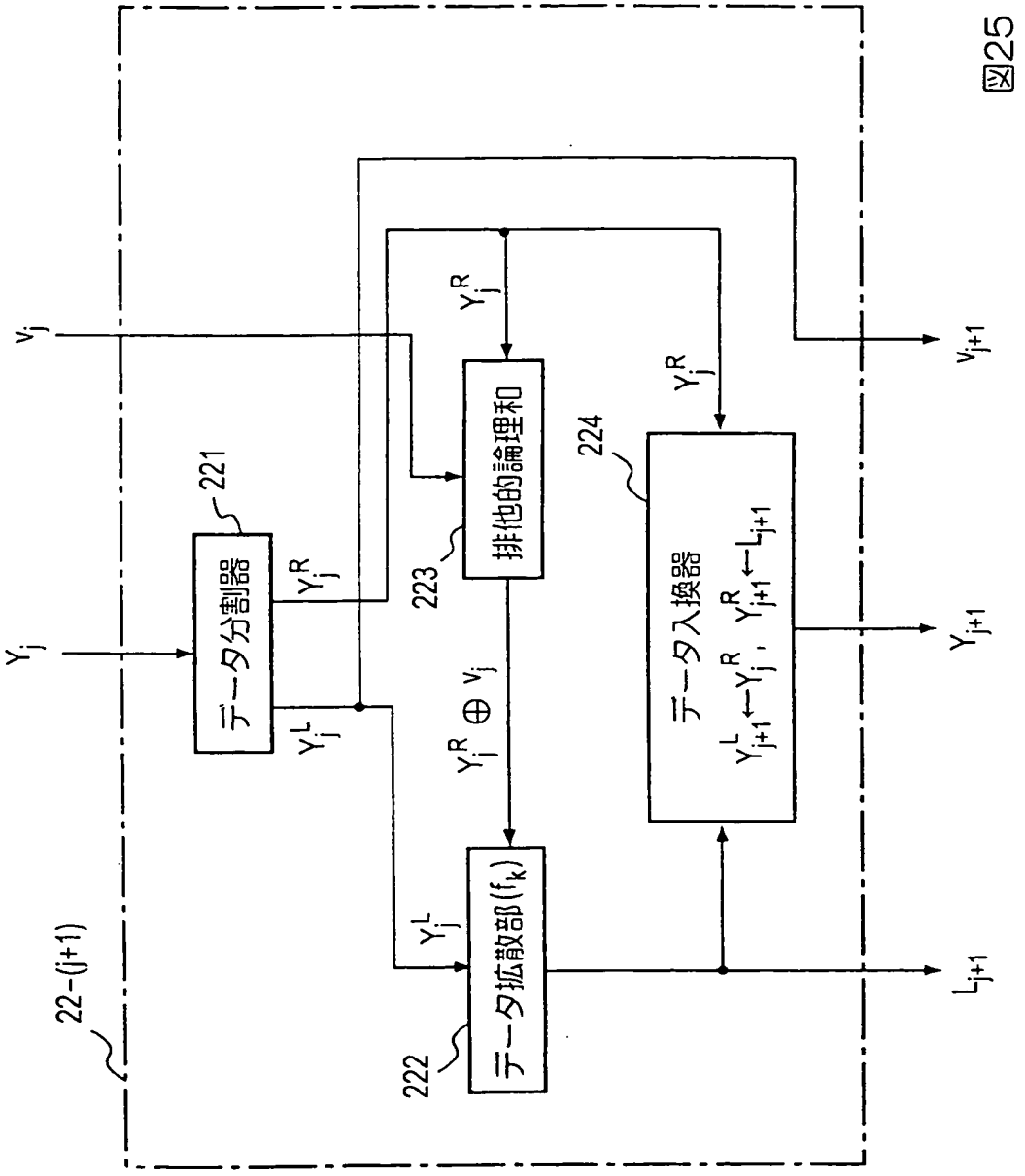

V_j
Y_j
データ分割器
221
Y_j^R
Y_j^L
排他的論理和
223
$Y_j^R \oplus V_j$
データ拡散部(f_k)
222
Y_j^L
22-(j+1)
Y_j^R
データ入換器
224
$Y_{j+1}^L \leftarrow Y_j^R$, $Y_{j+1}^R \leftarrow L_{j+1}$
V_{j+1}
Y_{j+1}
L_{j+1}

図25

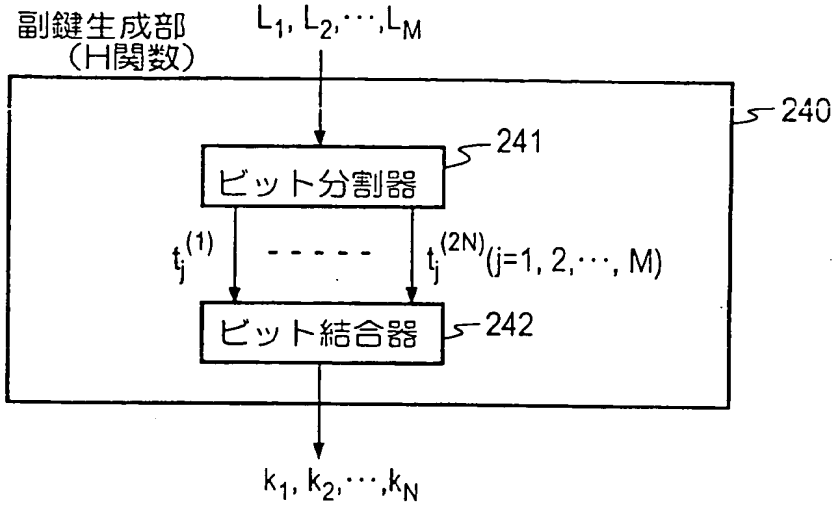

副鍵生成部
(H関数)
$L_1, L_2, \cdots, L_M$
240
241
ビット分割器
$t_j^{(1)}$ - - - - - $t_j^{(2N)}$ (j=1, 2, ···, M)
ビット結合器
242
$k_1, k_2, \cdots, k_N$

図26

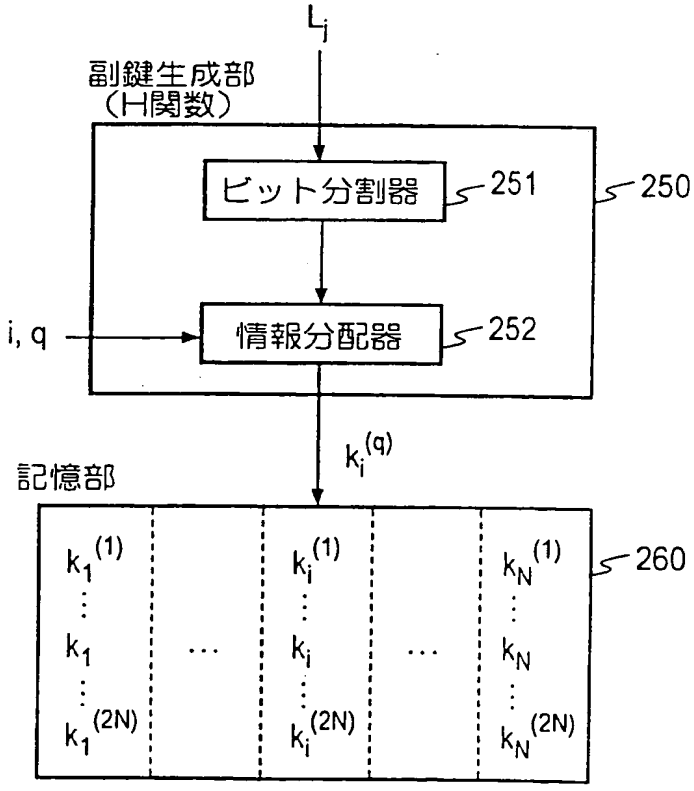

L_j
副鍵生成部
(H関数)
ビット分割器
251
250
i, q
情報分配器
252
$k_i^{(q)}$
記憶部
$k_1^{(1)}$ ⋮ k_1 ⋮ $k_1^{(2N)}$
…
$k_i^{(1)}$ ⋮ k_i ⋮ $k_i^{(2N)}$
…
$k_N^{(1)}$ ⋮ k_N ⋮ $k_N^{(2N)}$
260

図27

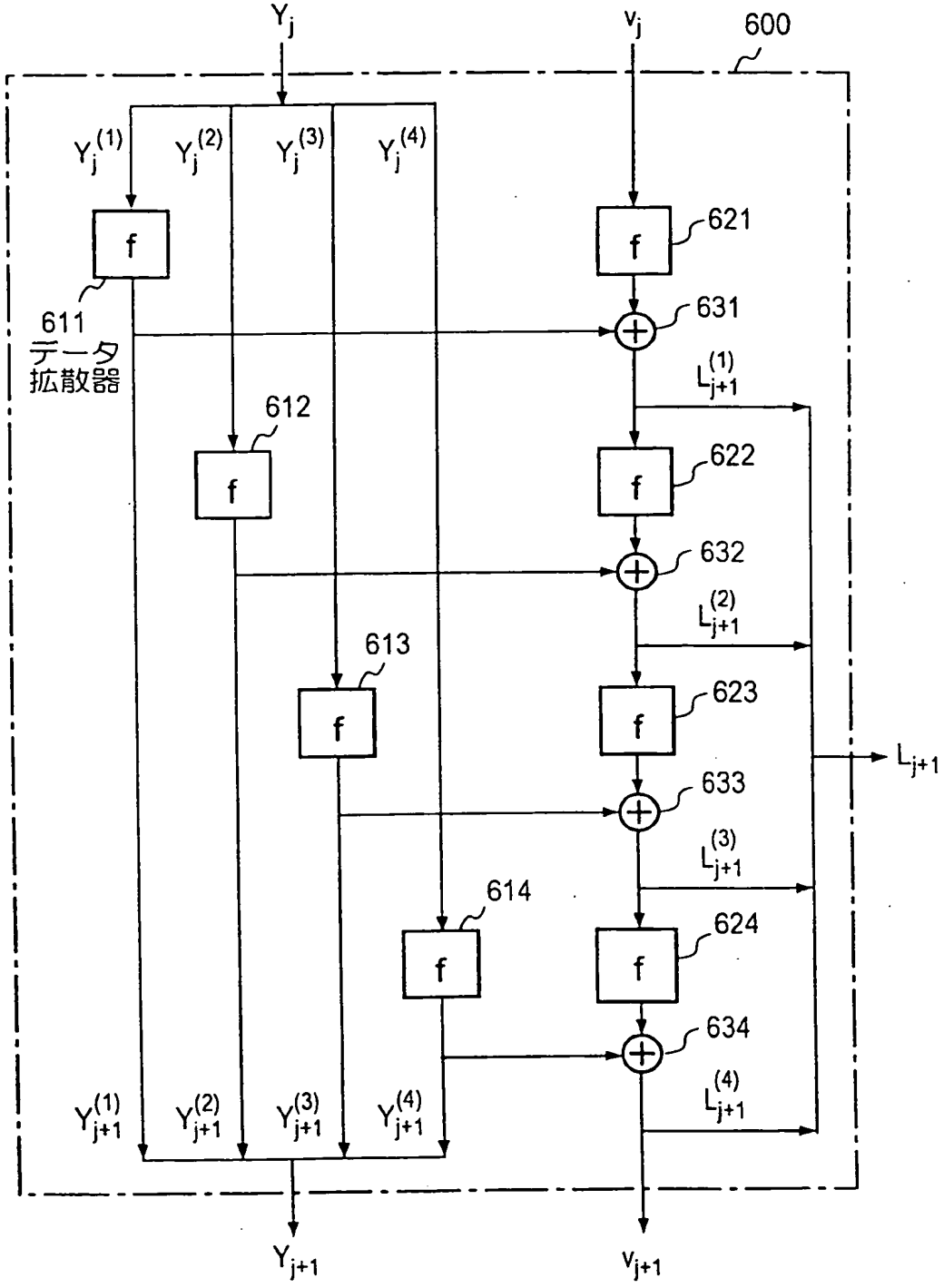

Yj
vj
600
Yj(1)
Yj(2)
Yj(3)
Yj(4)
f
611
データ
拡散器
612
613
614
621
622
623
624
631
632
633
634
Lj+1(1)
Lj+1(2)
Lj+1(3)
Lj+1(4)
Lj+1
Yj+1(1)
Yj+1(2)
Yj+1(3)
Yj+1(4)
Yj+1
vj+1

図28

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP99/00337

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[6] G09C1/00, H04L9/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl[6] G09C1/00-5/00, H04K1/00-3/00, H04L9/00-9/38

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
JICST File (JOIS), INSPEC (DIALOG)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	M. Kanda, S, Moriai, K. Aoki, H. Ueda, M. Ohkubo, Y. Takashima, K. Ohta and T. Matsumoto, "A New 128-bit Block Cipher E2," Technical Report of IEICE, Vol. 98, No. 227, (30 Sep 1998), p.13-24 (ISEC98-12) (in Japanese)	1-54
PA	M. Kanda, Y. Takashima and T. Matsumoto, "A round function structure consisting of few S-boxes (Part II)," The 1998 Symposium on Cryptography and Information Security, (28 Jan 1998), 2.2.D, (in Japanese)	1-46
PA	M. Kanda, Y. Takashima, T. Matsumoto, K. Aoki and K. Ohta, "A round function structure consisting of few s-boxes (Part III), " Technical Report of IEICE, Vol. 98, No. 48, (15 May 1998), p.21-30 (ISEC98-3) (in Japanese)	1-46
A	V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers and E. De Win, "The Cipher SHARK," Lecture Notes in Computer Science, Vol. 1039, (1996), p.99-111	1-54

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search
15 April, 1999 (15. 04. 99)

Date of mailing of the international search report
27 April, 1999 (27. 04. 99)

Name and mailing address of the ISA/
Japanese Patent Office

Facsimile No.

Authorized officer

Telephone No.

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP99/00337

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	M. Matsui, "New Structure of Block Ciphers with Provable Security against Differential and Linear Cryptanalysis," Lecture Notes in Computer Science, Vol. 1039, (1996), p.205-218	1-54
A	A. Shimizu and S. Miyaguchi, "Fast Data Encipherment Algorithm FEAL," Lecture Notes in Computer Science, Vol. 304, (1987), p.267-278	1-54

国際調査報告

国際出願番号　PCT／JP99／00337

A.　発明の属する分野の分類（国際特許分類（IPC））

Int. Cl⁶　G09C　1／00
　　H04L　9／06

B.　調査を行った分野

調査を行った最小限資料（国際特許分類（IPC））

Int. Cl⁶	G09C	1／00	－	5／00
	H04K	1／00	－	3／00
	H04L	9／00	－	9／38

最小限資料以外の資料で調査を行った分野に含まれるもの

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）
JICSTファイル（JOIS）
INSPEC（DIALOG）

C.　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
PX	M. Kanda, S. Moriai, K. Aoki, H. Ueda, M. Ohkubo, Y. Takashima, K. Ohta and T. Matsumoto, "A New 128-bit Block Cipher E2," Technical Report of IEICE, Vol. 98, No. 227, (30 Sep 1998), p. 13-24 (ISEC98-12) (in Japanese)	1-54
PA	M. Kanda, Y. Takashima and T. Matsumoto, "A round function structure consisting of few S-boxes (Part II)," The 1998 Symposium on Cryptography and Information Security, (28 Jan 1998), 2.2.D, (in Japanese)	1-46

☒ C欄の続きにも文献が列挙されている。　☐ パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー
「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」同一パテントファミリー文献

国際調査を完了した日 15. 04. 99	国際調査報告の発送日　27.04.99
国際調査機関の名称及びあて先 日本国特許庁（ISA／JP） 郵便番号100－8915 東京都千代田区霞が関三丁目4番3号	特許庁審査官（権限のある職員）　5W　9570 丸山　高政 電話番号　03－3581－1101　内線　6447

様式PCT／ISA／210（第2ページ）（1998年7月）

国際調査報告

国際出願番号 PCT／JP99／00337

C（続き）. 関連すると認められる文献

引用文献のカテゴリー＊	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
PA	M. Kanda, Y. Takashima, T. Matsumoto, K. Aoki and K. Ohta, "A round function structure consisting of few s-boxes (Part III)," Technical Report of IEICE, Vol. 98, No. 48, (15 May 1998), p. 21-30 (ISEC98-3) (in Japanese)	1-46
A	V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers and E. De Win, "The Cipher SHARK," Lecture Notes in Computer Science, Vol. 1039, (1996), p. 99-111	1-54
A	M. Matsui, "New Structure of Block Ciphers with Provable Security against Differential and Linear Cryptanalysis," Lecture Notes in Computer Science, Vol. 1039, (1996), p. 205-218	1-54
A	A. Shimizu and S. Miyaguchi, "Fast Data Encipherment Algorithm FEAL," Lecture Notes in Computer Science, Vol. 304, (1987), p. 267-278	1-54